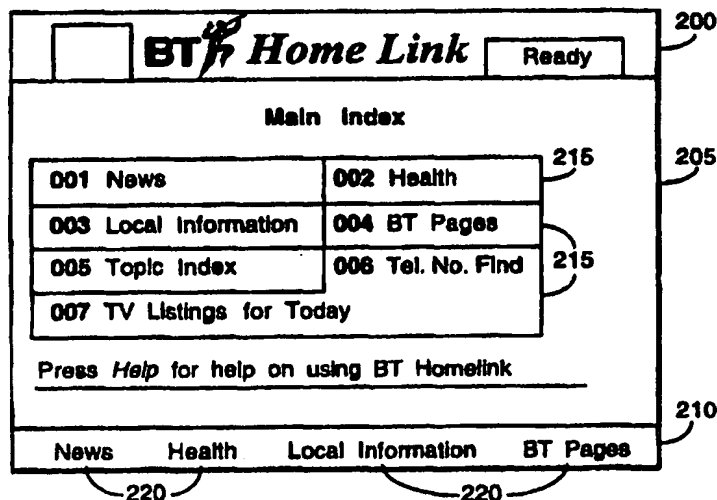




INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification ⁶ : G06F 17/30		(11) International Publication Number: WO 97/49044	
		(43) International Publication Date: 24 December 1997 (24.12.97)	
(21) International Application Number: PCT/GB97/01627		(74) Agent: DUTTON, Erica, Lindley, Graham; BT Group Legal Services, Intellectual Property Dept., Holborn Centre, 8th floor, 120 Holborn, London EC1N 2TE (GB).	
(22) International Filing Date: 17 June 1997 (17.06.97)			
(30) Priority Data: 9612627.1 17 June 1996 (17.06.96) GB 96304654.5 24 June 1996 (24.06.96) EP (34) Countries for which the regional or international application was filed: GB et al. 9707736.6 16 April 1997 (16.04.97) GB		(81) Designated States: AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, DE, DK, EE, ES, FI, GB, GE, GH, HU, IL, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, UA, UG, US, UZ, VN, YU, ZW, ARIPO patent (GH, KE, LS, MW, SD, SZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, ML, MR, NE, SN, TD, TG).	
(71) Applicant (for all designated States except US): BRITISH TELECOMMUNICATIONS PUBLIC LIMITED COMPANY [GB/GB]; 81 Newgate Street, London EC1A 7AJ (GB).		Published With international search report.	
(72) Inventors; and (75) Inventors/Applicants (for US only): EDWARDS, Nicholas, Hector [GB/GB]; 85 Leopold Road, Ipswich, Suffolk IP4 4RN (GB). RUSS, Martin [GB/GB]; 71 Farriers Close, Martlesham Heath, Ipswich, Suffolk, IP5 3SN (GB). GARNER, Paul [GB/GB]; 62 Mildmay Road, Ipswich, Suffolk IP3 1GG (GB).			

(54) Title: NETWORK BASED ACCESS SYSTEM



(57) Abstract

An interface (105, 125, 120) for a World Wide Web (WWW) (110) browser is described which recognises HyperText Markup Language (HTML) links embedded in WWW pages. Normally, such links are highlighted on a graphics screen and are activated using a pointing device such as a computer mouse. When each HTML link is recognised by the interface (105, 125, 120), it is assigned an identifier which can be selected by a user of a system incorporating the interface by means defined by the interface other than a mouse, for example by a keypad (105).

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	ES	Spain	LS	Lesotho	SI	Slovenia
AM	Armenia	FI	Finland	LT	Lithuania	SK	Slovakia
AT	Austria	FR	France	LU	Luxembourg	SN	Senegal
AU	Australia	GA	Gabon	LV	Latvia	SZ	Swaziland
AZ	Azerbaijan	GB	United Kingdom	MC	Monaco	TD	Chad
BA	Bosnia and Herzegovina	GE	Georgia	MD	Republic of Moldova	TG	Togo
BB	Barbados	GH	Ghana	MG	Madagascar	TJ	Tajikistan
BE	Belgium	GN	Guinea	MK	The former Yugoslav Republic of Macedonia	TM	Turkmenistan
BF	Burkina Faso	GR	Greece			TR	Turkey
BG	Bulgaria	HU	Hungary	ML	Mali	TT	Trinidad and Tobago
BJ	Benin	IE	Ireland	MN	Mongolia	UA	Ukraine
BR	Brazil	IL	Israel	MR	Mauritania	UG	Uganda
BY	Belarus	IS	Iceland	MW	Malawi	US	United States of America
CA	Canada	IT	Italy	MX	Mexico	UZ	Uzbekistan
CF	Central African Republic	JP	Japan	NE	Niger	VN	Viet Nam
CG	Congo	KE	Kenya	NL	Netherlands	YU	Yugoslavia
CH	Switzerland	KG	Kyrgyzstan	NO	Norway	ZW	Zimbabwe
CI	Côte d'Ivoire	KP	Democratic People's Republic of Korea	NZ	New Zealand		
CM	Cameroon			PL	Poland		
CN	China	KR	Republic of Korea	PT	Portugal		
CU	Cuba	KZ	Kazakhstan	RO	Romania		
CZ	Czech Republic	LC	Saint Lucia	RU	Russian Federation		
DE	Germany	LI	Liechtenstein	SD	Sudan		
DK	Denmark	LK	Sri Lanka	SE	Sweden		
EE	Estonia	LR	Liberia	SG	Singapore		

NETWORK BASED ACCESS SYSTEM

- 5 This invention relates to network based access systems in which access is provided by activating links in files containing addresses, for instance in the manner of activating links embedded in files written in HyperText Markup Language (HTML).
- 10 An increasing amount of information is becoming available on distributed communications systems. Most well-known perhaps is the World Wide Web (WWW) area of the Internet, accessed using WWW browsers. Pages consisting of text, graphics, audio files, video files etc. are each associated with an address in a distributed network by means of which they can be accessed. A review and
- 15 tutorial on HTML, particularly its use in the Web environment of the Internet, is published in the paper entitled "Creating a Hypertext Markup Language Document for an Information Server" by JYM Chu, WL Palva and DE Walter in Behaviour Research Methods, Instruments and Computers, 1995, Vol 27(2), at pages 200-205.
- 20 Pages, usually of text, are displayed on a screen. At system startup, the page displayed will usually be provided by a user's browser. Navigation from a first page to a location elsewhere in the network is achieved by means of a link embedded in the first page and visible on screen. Usually, the link holds the
- 25 address for the second location. When a user "clicks" on the link in the first page, the browser is activated to go to the location address held by the link. Thus the author of any page can make other pages, files or applications accessible regardless of their geographical location by links from their own page to relevant locations in the Internet.
- 30 (The term "page" as used herein should not be understood to refer only to pages of text and graphics but also to audio files, video files, Virtual Reality files and

computer applications (software) that may be made available via systems such as the Internet and WWW, if the context so indicates.)

Current WWW browsers such as Netscape use a click from a pointing device (for example a mouse) to select the links and thus to move between pages. That is, the device positions a cursor on the screen so as to identify a selected link. Although 'point and click' navigation is intuitive to people familiar with computer graphical user interfaces, this is not the case for a large proportion of the population. Furthermore there are many situations where a pointing device is not suitable for reasons of: cost, complexity, reliability, size, environment, etc.

According to the present invention, there is provided an access system for accessing a location in a network by activating a link in a file, which link contains a location address or an identifier for a location address, which system comprises:

15

i) means for retrieving one or more files;

ii) means for searching a file so retrieved to locate one or more links embedded therein;

20

iii) means for assigning an identifier to any link so located;

iv) selection means for use by the user to select an assigned identifier; and

25 v) activating means responsive to selection of an identifier to activate the associated link.

Depending on the nature of the link activated, or on the location associated with the link activated, the access system may then access a data file, such as text pages from the Internet, or may access other types of file, including videos or computer applications, such as conferencing applications.

The location associated with the link activated could alternatively be a location in a communications network, such as a ~~telephone or facsimile number or an electronic mail address~~. Equipment to enable a communications connection to be established in this way is disclosed in copending British patent applications GB9619958.3 and
5 GB9707712.7 filed on 29 September 1996, and 16 April 1997 respectively, in the name of the present Applicant.

~~By providing identifiers for the links, and providing a way of selecting a link via its identifier instead of operating directly, on-screen, on the link, embodiments of the~~
10 ~~present invention allow use of a different user interface from the pointing type of device, usually a mouse, usually used with such links. This means that the user can choose an interface which is better suited to them or more available to them than pointing devices such as a mouse.~~

15 It is even possible, using an embodiment of the present invention, to select a link which has been presented to the user non-visually, for instance by sound.

One example of an environment in which a different user interface is already utilised for other purposes is in the home. Remote control handsets are frequently
20 used to control domestic audiovisual equipment, for example: video tape recorders, hi-fi equipment and televisions. Also, a significant proportion of the television sets currently produced for use in the UK are equipped with Teletext which can only be accessed by using a remote control handset. This has established a familiar metaphor for controlling electronic equipment using a command-based control
25 system - based on key-presses rather than pointing. Teletext only functions with prepared information however. It cannot be used in a dynamic, uncontrolled environment such as with information from the Internet.

Embodiments of the present invention can be provided for instance as a terminal,
30 or the like, which can display pages from the World Wide Web, using an alternative "command-based" interface to select identifiers for links embedded in the pages seen on the terminal. For instance, there might be provided a simple Teletext-type remote control handset and simple coloured button-type labels for different links on

the screen. The user operates the handset to select a label, rather than having to use a 'point and click' WWW-type interface. Thus users do not have to position a pointer on the screen in order to select a path through a set of information pages. Instead navigation can be achieved by a series of key-presses which activate links.

5

It should be noted that, although the specific embodiment described below uses a television screen to display the WWW information, the 'command-based' interface is not restricted to use with a television screen: a wide range of other displays can be used. For example, a display panel on a telephone could be used to display the information, with the telephone keypad buttons being used to control the navigation, or a mobile information terminal could be produced, combining the functions of a mobile phone, pager, and personal organiser, whilst using a minimal set of buttons.

10 An example of the use of telephone keypad buttons being used in conjunction with a screen display is described in copending European patent application number 97300929.3, filed on 13th February 1997 by the present Applicant.

The physical implementation of the command interface need not be based on a key-pad on a remote control handset. There are a large number of alternative ways of providing user commands, and some of these are also detailed below.

An application which could be accessed by an embodiment of the present invention is a conferencing application. Screen-based audio-conferencing applications are described in copending British patent application numbers GB 9620000.1, filed on 25th September 1996, GB 9620260.1, filed on 27th September 1996, and GB 9705097.5, filed on 12th March 1997, and in copending European patent application number EP 97302615.6, filed on 16th April 1997, all in the name of the present applicant. By accessing a location at which an audio-conferencing application is accessible, the user would in fact most likely first be offered a text page for registration in the conference. Hence the launching of the conferencing application could be very similar to accessing a text page from the Internet.

An application to be launched is determined by a file extension of the link address. Another example of such an application is a video viewer which allows a user to view moving pictures. Many methods of coding moving pictures are available.

5 One widely adopted technique is that defined by International standard ISO-IEC11172 "Coding of Moving Pictures and Audio for Digital Storage Media at up to About 1.5Mbit/s", known as MPEG1. Files which conform to MPEG1 are conventionally named with a file extension of ".MPEG" or ".MPG". Files which are suitable for viewing with a QuickTimeTM viewer from AppleTM conventionally have

10 a file extension of ".MOV". If a link to such a file is located, an identifier is assigned to the link and an application is launched automatically if a user has specified the application to be associated with a particular file extension. Otherwise the user is asked to specify the application to be launched. Similarly a link may require a video conferencing application to be launched and a video

15 conferencing connection to be established. Such links need to include an application specific file extension for example ".VCL" for a videoconferencing link.

~~A particularly advantageous aspect of the present invention is that difficulties which would normally be met in presenting Web-compatible pages on a screen~~

20 ~~not normally for that purpose can be overcome.~~ As mentioned above, the links in a Web-compatible page are designed to be selected by the user using a pointing device to place a cursor at the position of the link in the text. Without a pointing device, it is not possible to select the link. However, by assigning a more generic identifier to the link, it becomes possible to use for instance a keypad.

25

~~The identifier assigned to a link may be any of several types of identifier.~~ It needs to identify each link as different from other links viewable at the same time on screen and it needs to be selectable by a non-pointing device such as a keypad. Hence, examples of identifiers assigned to links could include ~~colours, each link~~

30 being shown on screen in a different colour. Coloured buttons on a keypad could then be used to select a link. Just the initial letter of a link could be coloured. Another example is a number. Each link could be shown on screen with a number added and a keypad could be used to enter the relevant number.

Numbers lend themselves to use where a link is presented to the user as sound instead of visually. The existing link could have a spoken number added to it and the user would use that number as the link identifier.

5

~~The identifier for a link can also be at least partly derived from the normal on-screen appearance of the~~ link. For instance, the identifiers might be truncated versions of the text (or graphics) of the link itself. These identifiers might appear on screen in addition to the link itself. Hence the links may appear embedded in
10 the text on screen as is usual with Web pages, but there may also be provided an area on screen which shows just the identifiers, for ease of selection. In order to relate the identifiers to the links, it is useful that the identifiers include a truncated version of the text (or a symbol perhaps) from the relevant links. Embodiments of the present invention provide a means to generate that form of identifier at the
15 client end, thus allowing any original text having embedded navigation links to be used with such embodiments of the present invention.

In more detail, although it is possible for the author of a page incorporating a link to design the link specifically so that it could be used by embodiments of the
20 present invention, in Internet environments that would clearly have to be done in the page stored at the server to be accessed. It would be impractical in the usual Web-type environment to control the page design at all accessible servers. In order to make the system practical for all accessible pages, files, applications etc over the Internet, embodiments of the present invention are based on a client
25 environment which controls the way the links are shown at the client end. That is, a client device is provided which reads retrieved links and converts them wherever necessary by assigning an identifier which may in practice replace or supplement the link as it was originally authored, for display at the client device.

30 As indicated above, it may be preferred, for clarity, that embodiments of the present invention display not only the links as they appear in a page of text, but also the identifiers, displayed in a dedicated area of the screen. These identifiers may be provided by a set of "hot buttons" which repeat the functionality of the

links. For instance, it might be clearer to the user to have an array of buttons to select from, in an area across the bottom of the screen. This would be particularly so for a user familiar with the Teletext environment. Small display screens may not have the space available to show the page of text as well as an array of "hot buttons" having the originally authored on-screen appearance of the links. Embodiments of the present invention can then be particularly useful by replacing lengthy link descriptors with simple identifiers, such as numbers or colours, at least for an array of "hot buttons" added to the page of text.

10 Embodiments of the present invention are useful in environments where the size of the text is large relative to the screen area available for display, but the user is still going to need to select links. This occurs, as mentioned above, in cases where the screen display is simply small, such as in an aircraft where individual screens are provided for users. It also occurs however where it is required to provide large
15 text, for instance because the user has reduced visual ability or because the screen has to be visible from a distance. Embodiments of the present invention are generally useful in the manner in which link on-screen appearance can be tailored for the user environment without loss of functionality.

20 In general, embodiments of the present invention can provide a browsing capability, providing functionality for retrieving data and acting interactively as in known Internet browsers, with the added aspect of dynamic pre-processing of information at the user interface.

25 A terminal and interface system, herein referred to as "the Easy Terminal", will now be described as an embodiment of the present invention, by way of example only, with reference to the accompanying Figures in which:

Figure 1 shows schematically the main components of the Easy Terminal;

30 Figure 2 shows an example of a welcome screen;

Figure 3 shows an example layout for a remote control handset for use in the Easy Terminal;

Figure 4 shows a block diagram of the Easy Terminal hardware;

Figure 5 shows a flowchart overview of the software processes of the Easy Terminal;

Figure 6 shows a flowchart of a process used to analyse HTML to find links to other pages;

- 5 Figure 7 shows a flowchart of a process for creating text for text buttons in a display for the Easy Terminal;

Figure 8 shows a flowchart of a further process for creating text for text buttons in a display for the Easy Terminal;

- 10 Figure 9 shows a flowchart of a process for recording history of pages accessed by means of an Easy Terminal;

Figure 10 shows a flow chart of a process for responding to user selections by means of the remote control handset of Figure 3;

Figure 11 shows a flow chart of a process for ordering links displayed on a display by Easy Terminal; and

- 15 Figures 12 to 14 show flow charts of a process for clever scrolling of a screen display in Easy Terminal.

As described above, the Easy Terminal provides a simple information interface. The system is designed to provide an interface to electronic information in the
20 form of pages of text and graphics, audio files, video files etc. which can be accessed using key-presses. The information is derived from the WWW and is simplified and processed before being displayed.

Referring to Figure 1, the main components of the Easy Terminal are a display 100,
25 controlled by the user from a remote control handset 105. Information is accessed on the WWW 110 via a network connection 115, and processed by the Easy Terminal process software 120.

Effectively, Easy Terminal provides a Web browser of known type but with a
30 significantly different user interface providing significantly different control functionality.

The display unit 100 is a standard television. In order to ensure maximum legibility of text on the screen, large sans-serif fonts are used as much as possible. Anti-aliased fonts should be used if they are available. Text and background colours are chosen to ensure good contrast.

5

In addition to proportionately spaced text as described above, the screen can display high resolution colour images.

Referring to Figure 2, on switch-on, the user is presented with a welcome screen
10 with a general layout divided into three sections:

- Masthead 200
- Body 205
- Footer 210

15 The Masthead 200: the upper part of the screen acts as a fixed "masthead" which is used to identify the name of the service being provided, as well as provide feedback to the user. For instance, the number entered from the key-pad would appear at the upper left in this example, whilst the status appears at the upper right ('Ready').

20

The Body 205: the main portion of the screen is devoted to displaying WWW pages and/or processed WWW information. This part of the screen can be scrolled so that viewed pages can be larger than the available screen area. (The masthead and footer are not affected by this scrolling.) In general, pages which are
25 displayed are made up of five major elements:

- Headings which are normally large and in bold type. In this case, "Main Index" is shown.
- Text
- Links 215 (normally shown as highlighted text with an optional preceding three
30 digit number
- Graphics (pictures)
- Tables

Some pages, for instance home pages for particular services, may be simply a list of links from which the user can select further pages to access. The further pages will then generally comprise a mix of text and/or graphics and tables, with links embedded at irregular positions. The example shown in Figure 2 is of the home page type, showing simply a list of links.

The Footer 210: the lower part of the screen is also fixed, and provides space for four coloured text labels 220 - the colours are matched to the four colours of the buttons provided on the user's key-pad: Red, Green, Yellow, Blue. These are shortened versions of links 215 appearing in the body of the display 100 and are labelled with either descriptive text for the links 215 or a three digit number. The labels 220 enable any one of the links 215 in the body 205 of the display 100 to be activated. The labels 220 appear in a group of four and initially relate to the first four links 215 shown in the body 205 of the display. As the links shown in the body of the display are scrolled through, the labels can also effectively be scrolled through by activating a "Next" key on the keypad 105 which steps the labels on to show the next set of four links. Hence, if the labels display descriptive text or a link number, this can be changed as the links are scrolled. When all the links have been scrolled through and displayed, then the first set are re-displayed.

If descriptive text is used for the labelling, this can be derived from the link text shown in the body of the display 100.

Referring to Figure 3, the user may then access new WWW pages, which will take over the body of the display 100, using the remote control handset 105. The remote control unit 105 in this implementation is a television-type unit with a keypad. An example of a suitable handset has 25 buttons, as shown in Figure 3.

Twenty-five buttons represents a good compromise between the number of available functions and the numbers of buttons available on common remote control handsets.

Up and *Down* keys 300, 305 on the keypad scroll the current page seen in the body of the display 100 up and down. This can be done in a way which is referred to herein as "clever scrolling". The use of the *Up* and *Down* keys 300; 305 to move the on-screen view of the HTML page can be in two modes. The default mode is to provide scrolling which is related to the links on the page. When a page is initially displayed, the first four links are coloured to show their association with the four coloured buttons on the remote control or the on-screen coloured text label area. If the *Down* key is pressed, then the next four links on the page are coloured and the first four links revert to a default colour used to indicate links which are not available via the remote control or the on-screen coloured text label area. When the user presses the *Down* key, the system does a check as to whether there are links on the page which are not currently highlighted. When the last visible link on the page is coloured, then any subsequent pressing of the *Down* key will cause the screen to roll downwards (one line at a time) until either the end of the document is reached, or another link becomes visible which is not coloured. A similar mechanism can be used for the scrolling upwards process.

"Clever scrolling" in this manner is described in more detail below, with reference to Figures 12 to 14.

20

A *Back* button 310 can also be used for scrolling upwards in a minimal-button interface.

A *Next* button 315 can also be used to scroll the display - but this is linked to the number of links which are visible on the screen, and is intended for specialist applications where a minimal button-set is required. The *Next* button scrolls the body of the display downwards until four links are visible. These can then be activated by respective coloured text labels in the Footer of the display 100. Pressing the *Next* button then scrolls the display to present either the next page of text, or the next four links. Links 1, 5, 9 etc are thus always associated with the Red button, links 2, 6 and 10, etc the Green button, and so on.

30

In a non-minimal-button interface, the *Next* button is used to select the next four links on the page, regardless of their being displayed on the current screen - the coloured text labels would then update to reflect the change of links.

- 5 A *Summary* button 320 uses an online text summariser to produce a shortened form of the current page.

By pressing *Help* 325 and selecting *Settings* from a "user setup" page, the user may enlarge or reduce the size of the text, and select whether just text characters
10 or pictures and text characters are displayed.

A new page may be selected by one of two methods:

- Link and Number buttons
- Navigation buttons

15

The Link and Number buttons are the four coloured buttons 330, plus the numeric buttons 335. As mentioned above, the links in the body of the display 100 may have a preceding three digit number. To use the Link and Number buttons to select a link to activate, either one of the coloured buttons 330 can be pressed, or
20 a three digit code can be used, using the numeric buttons 335 on the keypad 105.

If one of the coloured buttons on the remote control is pressed, this will produce a new page corresponding to the appropriate coloured label 220 in the footer of the display 100. The label 220 will be showing a description identifying the link it is
25 currently related to in the body of the display 100 - it is this related link which will then be activated.

If alternatively a three digit code is entered, using the numeric keypad 105, as soon as the third button is pressed, then the associated link is activated and a new
30 page is retrieved. If a correction is required, the most recently entered digit may be deleted with the *Del* key.

There are many possible Navigation buttons which provide dedicated functions when they are pressed. Some of the possibilities include:

- Back* 310 allows the user to return to the page previously displayed
- 5 *Index* 340 displays the main index page which is seen at switch-on
- Help* 325 displays a menu of help pages on the use of the terminal
- Top10* 345 displays a list of pages which have been frequently and recently visited.

- 10 There are also three control keys which allow the user to control the loading of the pages:

- Stop* 350 Abandons the retrieval of a requested page. This may be used if the page is unavailable because of network problems or if the download is unacceptably slow

- 15 *Undo* (not shown) This button can be used to combine the functions of the *Stop* and the *Del* key - using the context at the time of use.

20

- Reload* 355 This refreshes the current page, downloading a new copy from the network. This may be used to ensure that the displayed page is up-to-date.

25

- Smaller numbers of buttons can be used by omitting features, or by requiring combinations of buttons to be pressed simultaneously. A minimal set of buttons would be approximately six buttons: the four coloured buttons, plus the 'Next' button, and a 'Back' button. The precise function of the named buttons is
- 30 described later.

An example of a minimal set of buttons is provided by the simple "game pad" controller - as used widely for home video/computer game consoles. The "game

pad" consists of eight buttons arranged in two groupings: four coloured buttons plus an additional four buttons conventionally used to indicate direction, rather like a primitive joystick.

5 The utilisation of the game pad buttons in an embodiment of the present invention is as follows:

- the four coloured buttons are used to select links on the page
- "Up" and "Down" buttons which scroll the page and the coloured links (using
10 "clever scrolling" as described below)
- a "Back/Delete" button. This functions as "back" except when the user is editing text, in which case it functions as "delete"
- an "Index/Finish" button. This takes the user to a main index page unless text is being edited, in which case it finishes the text

15

Other minimal button interfaces are also possible. The interface provision of embodiments of the present invention can be easily extended.

Simple keypads that provide only the digits 0 to 9, plus two additional buttons
20 (conventionally as a three column by four row block) can be used by redefining the function of the buttons on the keypad:

- the "0" button becomes the "Back" button
- the two additional buttons are used for paging upwards and downwards

25

Referring to Figure 4, the implementation of the Easy Terminal described here consists of a hardware platform and the associated software. The system is a linked mixture of hardware and software elements, and both parts are required to produce the display.

30

The hardware platform for the embodiment of the Easy Terminal described here consists of a number of components. Many of these would be common to most alternative implementations.

The components are detailed below:

Communications Network 110	An Ethernet based connection to the Internet
5 Network Link 115	An Ethernet communications card for a personal computer (PC)
Processor 120	A Gateway P4D-66 IBM Compatible PC Computer with 486 DX2/66 Processor.
Hard disk 43	A 540 MB Internal Hard Disk Drive
10 Infra-red Handset 105	A Propresenter Plus 25 button handset for PC
Infra-red receiver 41	A Propresenter URC receiver for PC
Graphics Card 44	An ATI MACH 64 PCI graphics card
TV Output 45	Derived using a Creative Laboratories TV Coder for PC

15

The hardware is in four parts:

- Control
- Input
- Processing
- 20 • Output

The Control hardware comprises the Infra-red Handset 105 and Receiver 41.

The Input hardware comprises the Communications and Network Link 115, which
25 connects the PC to the WWW 110.

The Processor 120 itself carries out the Processing of the Control and Input data in order to produce the Output display.

30 The Output is in the form of a TV picture, produced from the computer display by using a graphics card and a specialised TV converter.

Figure 5 shows a flowchart showing an overview of the software and the processes carried out in the Easy Terminal. The code is written in Borland Delphi (Object Oriented Pascal) for Microsoft Windows 3.1. The primary steps are as set out in Figure 5:

5

- STEP 520 : Search HTML file for links to other pages
- STEP 545 : Fit coloured labels to screen and display
- STEP 550 : Record history
- STEP 555 : Display HTML

10

Referring to Figures 6, 7, 8 and 9, each of the primary steps 520, 545, 550, 555 shown in Figure 5 is now described in more detail.

Search HTML File for Links to other pages: STEP 520

15

An HTML file contains reference to other pages (also known as "links") in the following form:

Link Text

20

where the <a and are tags which indicate the beginning and end of the reference respectively. The page reference indicates the WWW address of the page indicated, and will be in the form "http://...". Note that those parts of the address which are the same as the present page are not always given, and that
25 some expansion of the address is often required to create the complete address required.

Referring to Figure 6, when a new page is to be displayed in the body of the display 100, it needs to be searched, **STEP 520**, for links to other pages so that
30 those links can be differently displayed from ordinary text in the body of the display 100, and so that the labels in the footer of the display 100 will display appropriate colours or descriptions.

The system will therefore start, **STEP 500**, for instance, when a new (unprocessed) page has been accessed, usually because it has been requested by a user.

- 5 **STEP 600:** The HTML file containing the page is opened and, **STEP 605**, the process will read each character of the text in turn to see whether it introduces a reference, or a link, to another page. To do that, it runs a test, **STEP 610**, "Does text read "< a' ?" If the answer is yes, the process will read the file through to the symbol "", **STEP 615**. Between these two symbols, the process takes the
- 10 text as a page reference, or link, **STEP 620**, and evaluates that link for its full path name, **STEP 625**. (Often HTML addresses are abbreviated to the parts of the address which are different from the current page address. It is necessary to expand these partial addresses so that they can be compared with stored addresses in the history lists etc). The system then adds the link to a list of the
- 15 links or page references for the page being searched, **STEP 630**, and stores the text of the page reference prior to processing for the labels, **STEP 635**.

In **STEP 640**, the system will add a three digit number to the front of the page reference, or link text, for display in the body of the display 100. This three digit

20 number is also added to the appropriate page reference in the list of links found in the page being searched and will appear in, or at least be correlated with, a label at any time that the relevant link is one of the first four links displayed in the body of the display 100.

- 25 The process will then search for further links, until the end of the file is reached, **STEP 645**, and the process is terminated, **STEP 650**.

Shortening Link Text: STEP 545

- 30 The Easy Terminal displays four coloured text labels at the bottom of the screen. Most Internet link descriptions are too long, and may require truncation in order that the total length of the four labels will fit onto the screen width in a font which is legible on the display.

For the television display used in this implementation, the link descriptions are thus processed in the following way prior to display in order to make them as intelligible as possible.

5

Referring to Figure 7, where the text from a page reference has been stored for processing to provide a coloured label, **STEP 635**, the next step is to process the text for each page reference. This process starts with a check, **STEPS 700, 705**, whether the page reference contains text to be truncated to provide a label or
10 whether it contains an indicator that a label has already been supplied by a user. If the label has already been supplied, then the system will use that label, **STEP 710**. The process stops for that particular page reference, moving on to start again for the next.

15 To allow an author (or other user) to provide the labels, a special-purpose HTML tag has been defined. This enables HTML authors to embed pre-prepared shortened text labels for the coloured text labels. An example would be:

```
<A HREF = "page reference" HOTKEY = "Link1">Link text</A>
```

20

Thus the functionality of the <A command in HTML is extended.

If the label has not been defined by a user, the process checks whether the page reference, or link, is a picture, **STEP 715**. If the page reference is indeed a picture,
25 then the process looks for ALT text, **STEP 720**. If the text exists, this is used for deriving a label, **STEP 725**. If the ALT text is not there, then the system allocates a three digit number to the page reference concerned and uses this for the label, **STEP 730**.

30 Where the page reference was text, or ALT text existed, then the process starts truncating it. It removes any HTML commands, **STEP 735**. The page reference then has to be shortened.

First each link description is shortened to a length that represents the special case of four numbered links plus one text link, **STEP 740**. For the television example described here this is approximately 30 characters. Partial words left hanging at the end of a link are removed, **STEP 745**, and the link reduced by one character, **STEP 750**. Some unnecessary words such as "THE" and "A" may be removed to reduce the length of the link text still further, **STEP 755**. Words such as "AND" and "OR" can be shortened to "&" and "/", **STEP 760**. This process is a simplified form of the technique known as stemming, and more sophisticated processing may be used to improve the truncation efficiency.

10

Lastly, a number of non-standard HTML codes need to be converted before display, or removed, **STEP 765**. For example, a copyright symbol is written as ©, and needs to be converted to (C) before display, whilst the emboldening command and its reverse need to be removed.

15

After this first truncation step, the labels, or "button information", are stored.

Referring to Figure 8, even after these modifications have been made, the combined link text from the four link labels may still be too long, and will overspill the width of the screen. When the relevant page is to be displayed, the system will then go into a second truncation process. At this stage, the labels are processed in the fours in which they will appear on the screen during display.

The second truncation process makes a check step on the length of the labels, **STEP 800**. If the labels in combination are wider than the screen, the maximum length of the link labels is reduced by one character (ie first to 29, etc) with the longest link label being processed first, **STEP 805**. This 'balances' the length of the label text between the links. Partial words at the end of the text are removed, **STEP 810**. Finally, words like "AND" and "OR" are removed if they are the last word in the label text, and therefore redundant, **STEP 815**. The reduction of the length is continued until the labels will fit on the screen, and then they are displayed, **STEP 820**.

Recording History : STEP 550

Referring to Figure 9, a history of the pages selected by the user is important for a number of reasons:

5

- 1) the user may wish to back-track to a page they have recently looked at
- 2) the user may select a list of the 10 most recently visited pages
- 3) the user may wish to see a list of the 10 pages which they frequently visit
- 4) a log of the activity may be kept

10

Most of these functions are relatively simple. However 3) listed above requires more sophistication.

In the current implementation the Easy Terminal maintains a list of the 10 pages most frequently visited in the last 7 days of operation. In order to do this it must

15

first have a list of all pages which have been visited in the last 7 days: a history log containing counters, dates and addresses. The history log associates each page with seven day-counters which count the number of accesses of each page during that day.

20

Referring to Figures 5 and 9, when a new page has been accessed by a user, the page reference is added to an unsorted log file, **STEP 900**. This is a simple list of all pages accessed and can be effectively of unlimited length. The process of appending a new page reference is straightforward and not further described herein.

25

The new page reference is also added to a recently visited pages list, **STEP 905**. The recently visited pages list is maintained at a length of 10 page references by running a check each time a page reference is added, whether the list has gone over 10, **STEP 910**. If it has, the oldest page reference is dropped, **STEP 915**.

30

Again, this list is unsorted.

As well as the unsorted log file, an alphabetic list of the names of all pages visited is maintained, together with a count of the number of times a page has been

visited. To update the alphabetic list, a bisection search is carried out to find out whether the page has already got an entry, **STEP 920**. If no entry is present, then a new entry is created along with a new set of seven counters, **STEP 925**. If an entry is already present, then the relevant day-counter is incremented, **STEP 930**.

5

The alphabetic list allows the user to review which pages have been most frequently visited. A button on a keypad 105 or a three digit code, can be allocated to a function "Show 10 most frequently visited pages". If this is selected, the system can sort the alphabetic list according to the contents of the
10 counters. Thus when a list of the 10 most frequently visited pages is selected, the system counts up the total number of times each of the pages in the list has been accessed over the last 7 days, and sorts the list so that the top ten entries can be displayed.

15 Each time a page is accessed, the counter for the current day is incremented. When the beginning of a new day is detected, the counters are updated, and pages which have not been accessed over the last 7 days are removed from the alphabetic list.

20 Other schemes for maintaining a "Top 10" list are of course possible. Some possibilities are suggested below, in discussion of alternative embodiments and possible refinements in the present invention.

25

Displaying HTML : STEP 555

Commercially available HTML displaying software is used to display the modified
30 HTML code. This is not therefore described further herein. A number of minor modifications need to be made to the code however to remove error messages when images are not displayed and to remove the underlining from HTML hotlinks.

Referring to Figure 10, the remote control handset 105 sends commands using a standard serial port protocol. When a button is pressed, a Windows event is generated which activates a subroutine with the function shown in Figure 10. The user interface is thus a combination of the screen display 100 (specifically the
5 coloured text labels) and the coloured buttons on the remote control 105.

When the button is pressed on the remote control 105, this activates the subroutine to retrieve the data incoming from the remote control 105 at the serial port, **STEP 1000**. The subroutine will then act on the data retrieved, firstly by
10 matching the data against possible button identities, **STEPS 1005, 1010, 1015, 1020, 1025, 1030, 1035**. Depending on the outcome of each check, the subroutine will kick off a different process. For instance, if the key is found to be a digit key, **STEP 1005**, the subroutine will add the relevant digit to a current number selection store, **STEP 1045**. If there are now three digits in the current
15 number selection store, this is sufficient to identify a page reference and the subroutine will translate the three digits to a page reference, **STEP 1050**, by reference to the link list maintained above at **STEPS 630 and 640**.

The subroutine is now enabled to fetch the page from the WWW.
20

If the key does not represent a digit but is a hotkey (that is, identifies a label), then the system will have sufficient information to go direct to the link list, **STEP 1050**, and fetch the page.

25 If the key provides any of the following functions, then the subroutine will fetch a page directly: "Index, Help, Top 10, Reload, Back or Summary", **STEPS 1015, 1020**.

The key may be a control key acting on the screen, such as "Down, Up, Undo",
30 **STEPS 1025, 1030**. The subroutine will then scroll or update the screen or undo the last key press appropriately.

The key may have been the key "next", for displaying the next four coloured text labels. In this case, the subroutine will recognise the "next" command, **STEP 1035**, and display the labels as requested, **STEP 1065**. To support the "Next" key function, the system has a counter which counts the number of times the Next key has been pressed while viewing a document. This enables the system to track which set of four labels should be being displayed.

The last option in this embodiment is that the key represents a "stop" command, **STEP 1040**. The subroutine will respond by abandoning loading of the next page, **STEP 1070**.

An advantageous feature for embodiments of the present invention is to determine the order of links appearing in a page on screen, and therefore to control colour and/or number allocation, according to the position of the links of the screen rather than their order in the text. This can ensure that links will always appear in sequence when the document is read in a conventional direction, for instance from left to right and top to bottom in a Western environment. Problems can occur otherwise for instance in tables which can have the effect that numbers appear out of order on the screen - for instance when only a few links are visible in a multi-column table - the first column of links may only display one or two coloured or numbered links whilst the remaining columns have no coloured or numbered links.

This can be resolved by the system noting the co-ordinates of the links on the screen in the current viewed page whilst the page is being pre-processed. The links can be assigned to numbers in sequence according to their "y" and then their "x" co-ordinates. Once this order has been established, then the colours or numbers can be applied. This process will only need to be repeated for the page if the presentation of the page in the window changes, for example if the font size is changed.

Figure 11 gives an example of how improved link ordering could be implemented and the following description should be read in conjunction therewith.

The HTML is read in item by item (**STEP 1100**) . An item is either a section of text, a code or object within the page. The size of each item is then calculated, and then 'placed' on the page with an X and Y coordinate relative to the top-left of the page. If the object is a link (**STEP 1105**), the space would be left for later insertion of the link number (**STEP 1110**). (If no improved link ordering was present, this link number would be a sequential number in the order in which the links were found in the HTML. Thus once the entire file had been read, the page could be drawn on screen).

10

However with improved link order, the X and Y co-ordinates of each link would be noted in integer arrays (**STEP 1115**) along with an index array N which would contain a sequentially assigned integer (L in the Figure) (**STEP 1120**).

15 Once the entire file has been read in, a ripple sort algorithm (**STEPS 1125-1155**) is used to find the link which is has the lowest Y co-ordinate, and is thus closest to the top of the page. If more than one link has the same Y co-ordinate, the links are ordered according to their X co-ordinate, so that the links read sequentially from left to right. This link is assigned to a number which is then increased by one. As
20 the process is repeated, the links are thus ordered. Eventually the links are ordered according to their Y and X co-ordinates. The array N acts as a reference to the action which must be taken in the event of the link being selected.

For example, suppose the HTML is as follows

25

```
<table> <tr>
<td> <a href = "Item1"> Apples> </a> <br>
<a href = "Item2"> Bananas> </a> </td>
<td> <a href = "Item3"> Pears> </a> <br>
30 <a href = "Item4"> Plums> </a> </td>
</table>
```

On a standard HTML browser, this will produce a matrix of 4 links

<u>Apples</u>	<u>Pears</u>
<u>Bananas</u>	<u>Plums</u>

- 5 Without Improved Link ordering, in Easy Terminal, the links are numbered according to the order in which the links appear in the HTML code, thus the links will appear as

	<u>001 Apples</u>	<u>003 Pears</u>
10	<u>002 Bananas</u>	<u>004 Plums</u>

However, with Improved Link Ordering, the links will be re-ordered as:

	<u>001 Apples</u>	<u>002 Pears</u>
15	<u>003 Bananas</u>	<u>004 Plums</u>

- and the array N will contain the elements (1,3,2,4). If the user now selects link 2, Easy Terminal can use the reference array N to see that the second element N(2) is 3, and thus the appropriate action is that of the third link in the HTML, i.e. to
20 reference "Item3".

Referring to Figures 12 to 14, Easy Terminal can provide "clever scrolling" as follows.

- 25 In "clever scrolling", it is possible to make the colour of all the text uniform (normally black), and then highlight the available links (associated with the coloured text labels) with the four colours. It is also possible to colour the unavailable links with a default colour (for example purple) so that the links can be identified on the screen. The Next/Down button can thus be considered as a
30 button which moves the coloured (red, green, yellow and blue) labelling to the next set of purple links, whilst the Back/Up button moves the colours to the previous set of purple links.

When a request is made to plot the current page in the available window on the screen, the index of the first 'active link' to which a coloured button is assigned is noted as A (STEP 1200). Then the location of the first object which is visible on the page is determined (STEP 1205). If this object is a link (STEP 1210), then the index of this link, L, is compared to A (STEP 1220). If L is less than A, then there are links present on the page which the user can access by pressing UP, and these links are coloured purple. A boolean flag BEFORE is set to TRUE (STEP 1225), indicating that there are links with an index less than A present on the page. Likewise if $L > A + 3$ then there are links available to the user by pressing DOWN, so these links are also coloured purple, but the boolean flag AFTER is set to TRUE (STEP 1230). Otherwise, the link is associated with one of the coloured buttons, and is assigned to be RED, GREEN, YELLOW, or BLUE (STEPS 1235). This process is repeated until all the items visible on the page have been plotted, and assigned to appropriate colours.

15

Referring to Figure 13, now if the user presses down and AFTER is TRUE (STEP 1300), then the active links are moved down the page by increasing A by 4 (STEP 1305). Otherwise, there are no further links on the page, and the page is therefore scrolled downwards if possible (STEP 1310).

20

Referring to Figure 14, similarly if the user presses up and BEFORE is TRUE (STEP 1400), then the active links are moved up the page by decreasing A by 4 (STEP 1405). Otherwise, there are no further links on the page, and the page is therefore scrolled upwards if possible (STEP 1410).

25

ALTERNATIVES AND MODIFICATIONS

30 There are many alternative ways in which Easy Terminal could be implemented, which would provide a similar level of functionality to the user but with modifications to the hardware and software described. These are described and discussed below.

Although the specific embodiment described herein uses an IBM Compatible PC with an Ethernet connection, there are many alternative platforms on which the Easy Terminal could be implemented. Furthermore, the reprocessing and repurposing of the WWW information could be carried out remotely, or in the network, which could simplify the design of the user terminal hardware. Such an arrangement could be particularly appropriate where there are multiple users of the same system, such as in an interactive screen environment on an aeroplane for passengers.

10

A videophone could be used as a display for the Easy Terminal. Key stroke commands could either be taken from the videophone keypad, via speech recognition, or from an external controller.

15 A pager, watch, mobile phone or other mobile device could be used as an Easy Terminal to display processed WWW information and using 'minimum button set' navigation techniques as described herein.

The Easy Terminal software could of course run on any computer platform which had sufficient speed, memory and display capabilities. This could be provided for instance by a video recorder, satellite broadcast receiver, digital broadcast decoder, digital video player or a games console.

The Easy Terminal could run on an appropriately adapted "interactive TV" or "Video on Demand" system. This could be implemented either by running the Easy Terminal at the user's premises or at the content provider's. Thus in the first case, the data transmitted across the connection would consist of a conventional Internet data stream, and in the second it would consist of either an encoded video signal or an embedded data stream.

30

An EasyTerminal could be built in to a television receiver in much the same way that Teletext is currently included in the design of television receivers. In this case

it would be possible to improve the quality of the display by using a non-interlaced high definition mode.

The Easy Terminal could be implemented in two parts: a standard Internet browser
5 at the user terminal modified to interpret commands with a remote control 105,
and a network based processor which modifies standard World Wide Web pages as
appropriate.

The Easy Terminal could use any appropriate communications medium for the
10 transfer of data. This includes, for example, a fixed or mobile telephone network,
a broadcast TV service or radio paging service.

The Easy Terminal could use any suitable remote control unit 105, or alternative
source of user commands. These could include:

- 15 • Larger controllers with alphanumeric keys
• Foot control pads, or controllers involving other parts of the human body
• Connections other than infra-red, including radio or wired links.
• User independent speech recognition could be used as a substitute for a key-
based controller - thus simple words such as 'red' and 'next' or 'twenty-three'
20 could provide the means of navigating between links.

In all of these alternatives, the principle of the Easy Terminal Navigation remains
the same: the user does not have to position a pointer in order to activate a
command. However, embodiments of the present invention do not exclude the
25 use of a pointing device. A mouse or a trackball can still be used for instance.
This enables the same user interface as a conventional browser but still brings
advantages related to the present invention. For instance, an on-screen bar used
for coloured text labels can be used as a way of quickly selecting links without the
need to move the mouse pointer to the specific location of the actual link itself.
30 This could be important to people whose control over the mouse is limited, eg by
physical impairment or by an environmental condition such as severe vibration.
The coloured text labels could be replaced by coloured boxes to make this
relationship explicit for those with impaired vision - which links into the particular

ability of embodiments of the present invention to display text at large font sizes. The mouse driver software could be adapted to restrict the mouse pointer position on the screen - limiting the movement to horizontal positions over the coloured text label bar for instance, or the position could be quantised to ease the selection
5 of the four boxes/areas/labels.

For situations where a mouse is not appropriate, simpler control devices like paddles or foot controllers could be used to provide the same control ability as a mouse.

10

The Easy Terminal software could be written in any appropriate computer language or protocol. This could be implemented as a plug in to standard software, or in an Internet language such as Java.

15 A number of the features of the Easy Terminal could be implemented as processes carried out remotely with respect to the user. For example, the insertion of numbered links and coloured hot links need not necessarily be implemented at the user's terminal. Such a processor could be provided as a network or broadcast service.

20

The Easy Terminal could also provide rapid access to conventional Teletext pages. These pages could be displayed either as facsimiles of the source pages, or could be enhanced using the display, linking and navigation techniques described here.

25 A facility could be provided for the input of text. This could be achieved in a number of ways. Some possibilities include:

- For any text field, the user could select from a number of options on the screen (this is known as a listbox or 'pop-out' box). The options which appear in the
30 box could include items already known from the setup of the Easy Terminal (the owner's name, address, telephone number etc.) as well as a history of recently entered text.

- The user could make any combination of letters up by selecting letters one by one from a scrolling or rotating list, or a grid of characters.
- Characters could be associated with the numeric keypad by pressing more than one button at once.
- 5 • Characters could be selected by pressing a button more than once. These characters could correspond to those currently written on telephone keypads. For example, to select the character 'B' the '2' button would be pressed twice since it has the legend 'ABC'.
- A full alphanumeric keypad could be included by having a larger number of
10 buttons on the keypad, or the option of attaching a standard keyboard to the terminal.

The summarising features which are already part of the specific embodiment described above could be extended in a number of ways.

- 15 • The user could be provided with the choice of a number of different levels of summary. This could be selected, for example, by repeated summary commands.
- Summarising of text could be carried from one page to the next. Thus, once the summariser is activated, it would be operational until a command is given to
20 disable it. This could be provided as an option or user preference.

The size of text used in HTML pages can be determined both by the author and the reader. The author of an HTML page can select the relative sizes of fonts with respect to a standard reference size; this can be varied throughout the document.

- 25 The reader may select a scaling factor to enlarge or reduce this reference size. A control could be provided to remove certain resizing information from the text so that fonts which are too large or too small are displayed in more appropriate sizes.

- 30 In addition, the ratio between the largest and smallest font sizes used on the display may be reduced in order to suit the type of display. For example, on a computer screen, headings in large font sizes are appropriate, whilst the same large headings may appear too large when viewed in association with blocks of text on a television screen.

Improved Navigation by Colours

- A control could be given to allow a user to set up a choice of standard colours for the background, the text and the links, or to use the colours and backgrounds as selected by the author of the page. A further refinement would allow only specific colour combinations to be allowed, whilst those which would significantly reduce the legibility of the page could be altered to maintain clarity.
- 10 As mentioned above, it is possible to use rectangular areas with coloured borders to indicate active areas. This can be extended to provide the option for coloured borders to picture elements (GIF graphics etc) on the screen. This enables links which are the equivalent of on-screen 'buttons' to be selected using colours.
- 15 Authors of HTML pages can suggest the background text and link colours for their pages. Certain colour combinations may be unsuitable for use with Easy Terminal's colour navigation schemes. One example of this might be the low contrast of a yellow link on a white page or a blue link on a black page. Easy Terminal overcomes this by adjusting the navigation colours according to the background selected. For example, a white background will cause the link colours to be darkened, etc. If a background colour is particularly close to one of the link colours, then Easy Terminal will darken the background and lighten the link colour until a reasonable level of contrast is provided.
- 20
- 25 Navigation between pages could be achieved using one or more of a number of possible alternative schemes:
- **Uniform Length Numbered Links**
- 30 Numbered links of a uniform length within a page (e.g. 01 02...99) could be used to allow users with a numeric keypad to select a new page. The page retrieval could start as soon as the final digit is pressed. Note that the number of digits need not necessarily be three, as described above, and could be varied

according to the number of links on the page. Thus a page with less than 10 links could use a single digit, whilst one with more than 10 would require two.

- **Variable Length Numbered Links**

5

Numbered links with a length which is variable within a page could be used. The user could indicate the end of a number either by pressing a return key or by having a set time-out after which the end of the entry would be assumed. For example, links might be enumerated as (1, 2, 3....10, 11, 12). In this
10 example, if a user pressed 1, there would be a time delay to allow the user to enter a second digit if required. On the other hand, if the user pressed 3, no time delay would be required as there are only 12 links on the page. By techniques such as these, users can navigate by means of numbers only and can remove the coloured text labels if required.

15

- **Coloured Text Labels displayed Separate to the Main Text Body**

As described above, coloured text labels which contain summaries of the link text on the page or specially written text can be displayed at the bottom of the
20 screen, and activated by means of associated coloured buttons on a keypad 105. The number of labels need not necessarily be four, of course, but could be varied according to the width of the screen. When more links are present on the screen than there are associated labels, one or two additional keys (*Next* in our specific embodiment) could be used to cycle through the available links
25 forward and/or backward.

Note that if the display used could not show text in different colours, coloured markers could be printed at the bottom of the screen close to the labels so that the association of the labels with the buttons is apparent.

30

- **Coloured Text Links within the Main Text Body**

In an alternative method of navigation, the labels need not carry any text at all, but just be different colours. This would involve removing the colour information from the main body of the text so that most of the text is displayed in one single colour (for example black). Links embedded in the text could then be displayed in different respective colours, associated with very simple coloured labels at the bottom of the screen. Thus, in the case of our specific embodiment, with four coloured labels, the first link in the text would appear red, the second green, and so on. One or two buttons on the keypad 105 could be provided to cycle through the links so that different links would be highlighted, ready for activation. Links in the text could indeed be coloured instead of the labels on the bottom of the page.

Although the specific embodiment described above uses alternative navigation schemes to access hypertext links within an HTML document, the same navigation scheme could be used to access HTML labels within forms or any other object on an HTML page.

Graphical image maps could be included in the navigation schemes by overlaying links on the images. Thus with a numbered navigation scheme, numbered links could be associated with areas of an image in many different ways. For example,

- the numbered link could flash over the active area of the image
- a number could be written beside the image with an arrow pointing to the active region

25

With a coloured navigation scheme, the active areas of the image could be highlighted by means of, for example,

- a coloured frame round the active area of the image
- changes to the colour palette information over a local area of an image, so that part of an image appears predominantly the colour of a link colour
- a flashing filled area of the appropriate colour could indicate a link.

30

Since the links within a map are available as a map file, then the links contained within an image could be presented as a series of sets of four coloured text labels, which are scrollable using the *Next* command. In this case, the only description available for the link could be the address contained within the link itself, and so
5 might contain useful information. One solution to this problem might be to pre-load the page pointed to by the link, and then use the title of that page as the text for the link label.

The number of keys used to control the system could be reduced or extended
10 according to the hardware platform and the user requirements. A number of examples of variation in key layouts might be:

- navigation between the pages could be done using either only numbers or only colours
- 15 • on scrolling the page up and down, as abovementioned the sets of four active links, ie with associated labels, could automatically be altered according to those currently visible on the screen. Thus the functions of *Next* and *Down* in the specific embodiment described above would be combined.
- the function of *Up* and *Back* in the specific embodiment described above could
20 be combined.

Audio could be used as a means of reinforcement. Thus different sounds could be played in response to key presses or commands.

25 Graphics display techniques could be used to make the operation of the Easy Terminal more intuitive and apparent. A number of examples are given below:

- A graphical indicator could be used to show the user how much and which part of a page is currently being displayed on a screen.
- 30 • The coloured text labels could be smoothly scrolled to indicate the relationship between different groups of labels. In the case of the specific embodiment described above, for example, upon pressing the *Next* button, the next four links would scroll into the visible part of the footer.

- Shading could be used to indicate that there are regions of the page which are not currently visible on screen.

For certain applications, access could be restricted to a subset of the Internet by using a set of specially designed pages which only contain links to other pages within the set. It would also be possible to remove families of links according to where they point. For example, all links could be removed which start with "http://undesirable.com".

10 Other Methods of Controlling Easy Terminal

'Back' key

- Selecting the 'O' button on a keypad can act as a synonym for the 'Back' key.
- 15 Pressing # and * (or additional buttons on the keypad) will scroll the current page up and down by the height of the current screen window.

Remote Control

- 20 Easy Terminal may be controlled by other applications. Messages are passed from the controlling application to Easy Terminal using either Windows messaging (DDE, OLE, ActiveX, COM, Java) technique, or hooking into the mouse or keyboard driver (by emulating key-presses, for example) on the same computer, or from other computer(s) or equipment via the serial, parallel or network ports. In one
- 25 implementation of Easy Terminal the network-based control is achieved by using TCP/IP. The use of Easy Terminal navigation with colours and numbers lends itself to remote control from other devices and to data-sharing applications because the controls and signals are not governed by screen layout.
- 30 In particular, the ability to user simple keyboard commands to provide control over the display of Web pages is not commonly available in other browsers - the control is normally assumed to be via the mouse. The selection of links is normally carried out by using the mouse to point to a link (normally underlined and coloured blue on

a conventional browser display) and then clicking the mouse button - in Easy Terminal this can be achieved by using either the coloured text labels, or the coloured remote control buttons (or their equivalent). Scrolling on a conventional browser required precise movement of the mouse to the scroll bar, and then
5 clicking or dragging, or else the use of the cursor keys on the keyboard. Easy Terminal's clever scrolling requires just the 'Next' or 'Back' buttons to achieve the same functions - and combines the on-screen highlighting of available links as well.

Easy Terminal thus allows keyboard control of a browser, which could be
10 important for any use where a mouse is inconvenient or reduces efficiency. One example of this would be in a situation where a person, whose typing skills are required for inputting information into a computer, is also required to use a web-based page. Easy Terminal enables this person to keep their hands on the keyboard and still control and interact with the screen display. This could be
15 particularly important in situations where time efficiency is paramount: Call Centres, Directory Enquiry and other 'bureau' based applications.

Linked Easy Terminals

20 Two Easy Terminal equipped computers can be linked together so that command and control messages can be transferred between them using one of the 'Remote Control' methods described above. This function is not currently available as a standard feature on other browsers. This allows the two Easy Terminals to be used for tutorial, education, illustration, form filling, and other instances where
25 interactivity between more than one user is required. This is only possible because of the simple control interface to the web browser which Easy Terminal provides. For example if two conventional browsers were linked together using application sharing, then control information about mouse position would form the majority of the transferred control messages, whereas for Easy Terminal a few button press
30 messages would be sufficient.

The types of information which can be exchanged are not restricted to just control messages. Additional information such as display settings, the current document

URL, highlighted link and position within the document, current frame may also be transferred.

Highlighted numbers

5

A number of known techniques may be used to highlight numbers on the page. This includes emboldening the number, displaying the number in "reverse video" (reversing number and background colour) or displaying the number with a different background colour.

10

Improved Page Presentation

Intelligent Masthead

- 15 The masthead which displays information to the user such as numbers pressed, the status of the current page or other instructions is only displayed where necessary. Thus when a page is fully loaded and is being displayed, the full area of the display is available.

20 Limited width pages

- Standard HTML browsers allow pages to have variable widths and heights. If either the width or height exceeds the screen parameters then the user is given the ability to scroll the document across the screen. Easy Terminal allows the user to
- 25 scroll the document up and down, but not left and right. In other words, the page is never allowed to exceed the display width. This is achieved in the following ways.

- Text is conveniently wrapped at spaces and line breaks such that it fills the
- 30 width of the screen. However if a single word is wider than the screen than the document must be scrolled to view the word. In Easy Terminal the font size of the word is reduced such that it fits in the available space.

- If an image is wider than the screen then it is scaled preserving its aspect ratio so that it fits the screen.
 - If a table is wider than the screen then its column widths are reduced such that the ratio of the desired widths remains constant. Text and images within a
- 5 table are scaled to fit the table cell if required.

Redundant Link removal

The HTML processor can consider two adjacent (successive) links on the same

10 page as being the same. Thus a picture and accompanying text (which both point to the same URL) are indicated as the same link by colour and/or number, for example.

Form Control Objects

15

Conventional browsers use on-screen control objects to provide user interaction with devices like buttons, checkboxes and listboxes - typically used in forms. User interaction with these typically involves mouse clicks. The on-screen control, size and colour of these objects is fixed by the operating system and cannot be easily

20 changed. When the font size used to display the text is changed, these objects do not scale to compensate, which means that, for large font sizes, they are disproportionately sized.

Easy Terminal can display standard HTML form control objects - but it displays

25 these by redrawing them from graphics primitives rather than using the standard operating system provision. This enables the size and other properties of these objects to be controlled. For example, each form control object can have a colour, a number and is scaled according to the size of the current font.

30 Typical control objects include:

Name	Function
button	submit a form or make a direct selection

	radiobutton	select only one of a number of options
	checkbox	select many options
	combobox	select one of a number of pop-up options
	listbox	selection one or more of a number of listed options
5	editbox	type in a single line alphanumeric string using a keyboard
	carousel	
	textarea	type in multiple alphanumeric strings

Some of the form control objects can be assigned to special functions such as a
10 control screen.

As with other Easy Terminal user interface features these control objects can be manipulated using numbers of colours. Some of these: the combobox, listbox, editbox and textarea require the user to select items within the control, again by
15 number or colour. In the case of the combobox and listbox, the items are numbered and coloured; in the edit box and text area, the carousel of letters is controlled via colour, and the control focus is shifted from the page to the control. Coloured links on the page temporarily revert to the default link colour whilst the focus is in the control object. The coloured text labels at the bottom of the screen
20 reflect the current colour selection options available to the user - in the case - the control object options.

Once the user has completed the selection within the control object, the focus returns to the page, and the link colouring will return.

25

Slideshows

A Easy Terminal Slideshow consists of a series of files or URLs and timing information. When the slideshow is selected the URLs are displayed in sequence
30 until interrupted by the user.

The slide show implementation of Easy Terminal uses files with the extension type.sho. These files consist of a list of times in seconds and URLs, for example:

10 URL1

5 5 URL2

In this example, URL1 will be shown for a period of 10 seconds, and URL2 will be shown for 5 seconds. After this, the cycle will be repeated until the user intervenes by pressing any of the control keys.

10

This functionality is attractive because no modifications are required to the pages in order for them to be used in a slide show.

This function can also be used with linked Easy Terminals where two users can
15 view the same slide show. The slide show stops when either user presses a control key.

20

25

30

CLAIMS

1. An access system for accessing a location in a network by activating a
5 link in a file, which link contains a location address or an identifier for a location address, which system comprises:
- i) means for retrieving one or more files;
 - 10 ii) means for searching a file so retrieved to locate one or more links embedded therein;
 - iii) means for assigning an identifier to any link so located; ✓
 - 15 iv) selection means for use by the user to select an assigned identifier; and ✓
 - v) activating means responsive to selection of an identifier to activate the ✓
associated link.
- 20 2. A system according to Claim 1, wherein the activating means activates the associated link so as to retrieve a file located at an address contained in or identified by that link.
3. A system according to Claim 1 or 2, wherein the activating means
25 activates the associated link so as to launch an application associated with the link.
4. A system according to Claim 1, 2 or 3, wherein the system further
comprises a display control output for use in displaying the file containing the link
30 to be activated.
5. A system according to Claim 4, wherein the system is further provided with a user input for display control commands, and control means to receive such

control commands and to control the display in accordance with said control commands.

6. A system according to any one of the preceding Claims, wherein the
5 system is provided with a translation data store for use in translating an identifier, when selected by use of said selection means, to an activatable link.

7. A system according to any one of Claims 4 to 6 wherein means is
provided to process at least one activatable link in a retrieved file, prior to display,
10 so as to modify the on-screen appearance of the link.

8. A system according to Claim 7 wherein the on-screen appearance of the
link is modified by the addition of the identifier assigned to that link.

15 9. A system according to Claim 8 wherein the identifier comprises a number.

10. A system according to either one of claims 8 or 9 wherein the identifier
comprises a colour.

20 11. A system according to any one of claims 4 to 10 wherein means is
provided to process at least one activatable link in a retrieved file so as to generate
a second on-screen appearance of the link.

12. A system according to Claim 11 wherein means is provided to process at
25 least two activatable links in a retrieved file so as to generate second on-screen
appearances of each respective link, and means is provided to display a set of said
second on-screen appearances of the links, separately from other content of said
retrieved file.

30 13. A system according to either one of Claims 11 or 12 wherein each said
second on-screen appearance(s) of a processed link is different from the first on-
screen appearance(s) of the associated processed link.

14. A system according to Claim 13 wherein said second on-screen appearance of a processed link comprises substantially only the identifier for that processed link.

5 15. A system according to any one of claims 12 to 14 wherein said set comprises a smaller number of second on-screen appearances of processed links than the number of links in a retrieved file.

10 16. A system according to Claim 15, said system being provided with link selection control means for use by the user to select the links in a retrieved file whose second on-screen appearances appear in said set.

17. A system according to Claim 16 wherein the system has means to respond to a scroll input, by means of the link selection control means, to scroll
15 the set of second on-screen appearances of links in relation to the links in a retrieved file.

18. A system according to Claim 17 wherein the means to respond to a scroll input comprises:

20

i) means to compare the currently displayed set of second on-screen appearances of links with the currently displayed first on-screen appearances of the links in a retrieved file; and

ii) means to scroll the portion of the retrieved file currently displayed

25

in the event that the last of said set and the last of the first on-screen appearances of the links, in the scroll direction, both relate to the same link.

19. A system according to any one of the preceding Claims wherein means is
30 provided to process at least one activatable link in a retrieved file, prior to display, so as to incorporate timing information, and wherein said activating means activates said processed link so as to display a file retrieved for a period of time determined by said timing information.

20. A system according to any one of the preceding claims wherein the selection means comprises a remote control device, for instance based on infra-red transmission.

5

21. A system according to any one of the preceding claims wherein the system is further provided with means for processing links, located by the means for searching a file, to generate an identifier for each link from information contained in the link, and means for displaying the file together with one or more
10 identifiers so generated.

22. A system according to Claim 21 wherein the means for processing links comprises a truncation device for generating a truncated version of a link.

15 23. A system according to any one of the preceding claims which further comprises means for preparing files with one or more embedded links and loading a prepared file to the system, at least one embedded link in a prepared file including, at least part of, an identifier to be assigned by the means for assigning identifiers, and further including an alert recognisable by the system to alert the system that
20 said embedded link includes such an identifier or part of an identifier.

24. A system according to any one of the preceding claims wherein the file is written in HyperText Markup Language.

25 25. A system according to any one of Claims 4 to 24, which further comprises means for determining spatial co-ordinates for links located in a file, said spatial co-ordinates relating each link to its location in the file when displayed, and the means for assigning an identifier assigns an identifier to each link in accordance with its spatial co-ordinates.

30

26. A system according to Claim 25 wherein said spatial co-ordinates are based on perpendicular axes and the means for assigning an identifier gives preference to the co-ordinate for one of the perpendicular axes, in assigning identifiers to a respective link.

27. A system according to any one of the preceding claims wherein the system further comprises display control means to control the on-screen appearance of a file, in addition to the on-screen appearance of one or more links
5 embedded in the file.

28. A system according to any one of the preceding claims wherein the activating means activates the associated link so as to set up a communication connection in one or more communication networks.

10

29. A system according to any one of the preceding claims, comprising a television display output for displaying a retrieved file.

30. An information network browser comprising a system according to any
15 one of the preceding claims.

31. An information network browser for locating, retrieving and displaying files stored at a location accessible by means of an information network, said browser comprising means to process a retrieved file prior to display.

20

32. A browser according to Claim 31 wherein said means to process a retrieved file comprises means to identify an activatable link embedded in said file, to generate an identifier for a link so identified, and to display the identifier together with the link, the browser further comprising selection means for use by
25 the user to input the identifier so as to activate the link.

33. A browser according to Claim 32 wherein the selection means comprises an array of independently selectable input means, each input means being dedicated to a respective identifier.

30

34. A browser according to Claim 33 wherein the selection means comprises a keypad.

34. A browser according to any one of claims 31 to 33, further comprising control means for use by the user to control the display.

Fig.1.

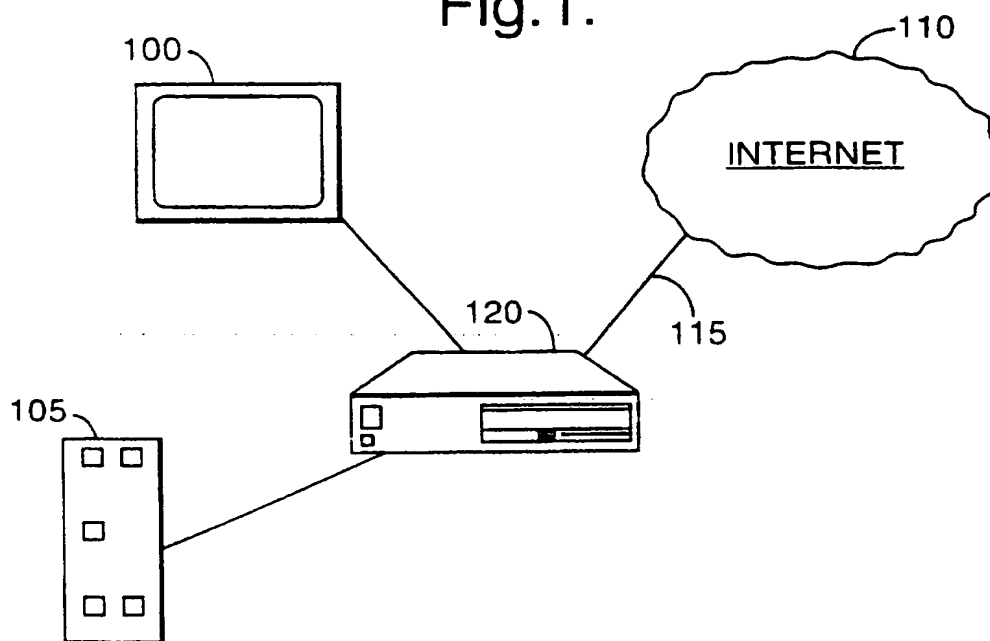
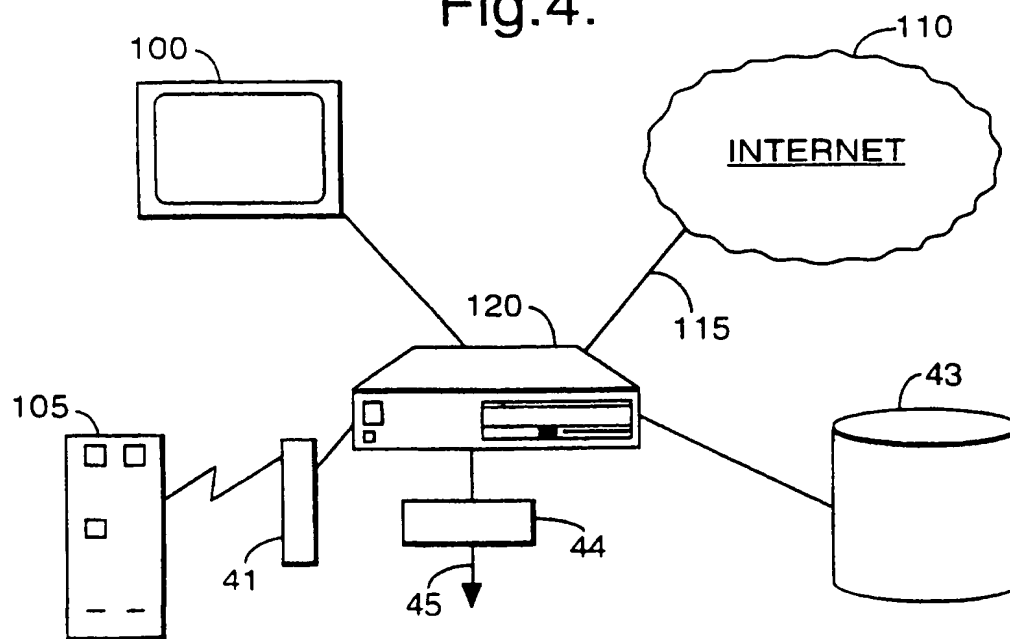


Fig.4.



2/10

Fig.2.

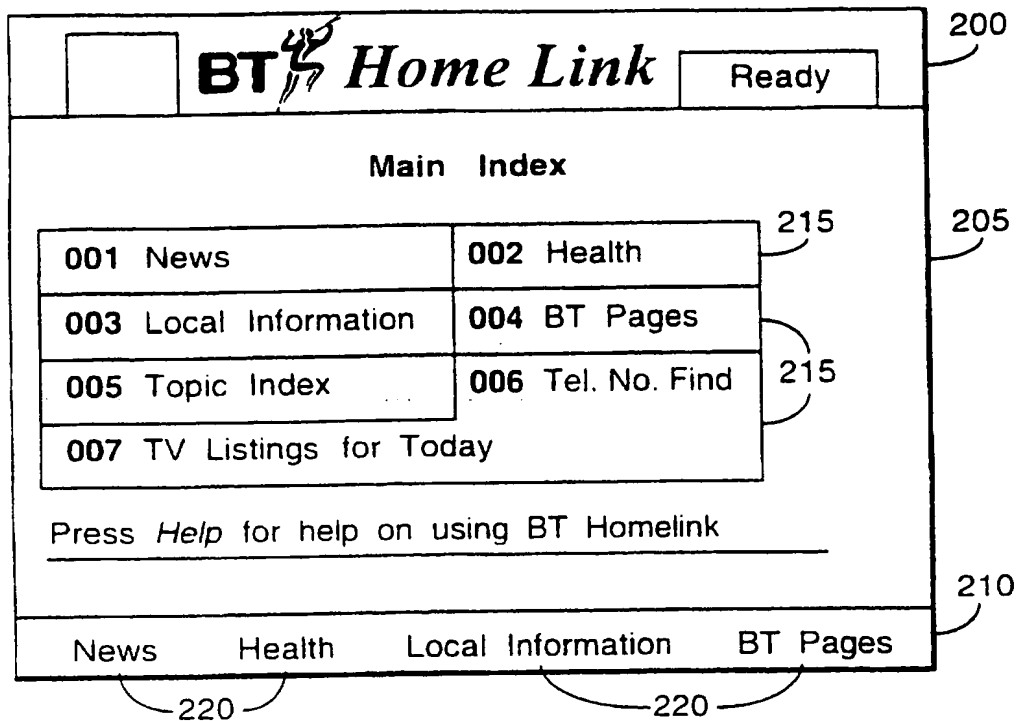


Fig.3.

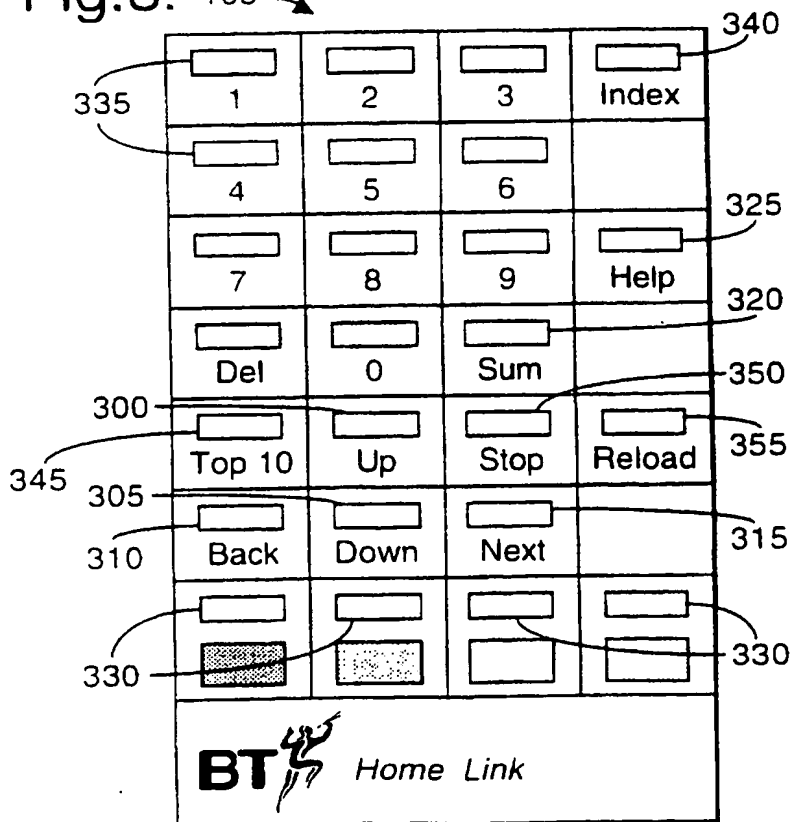


Fig.5.

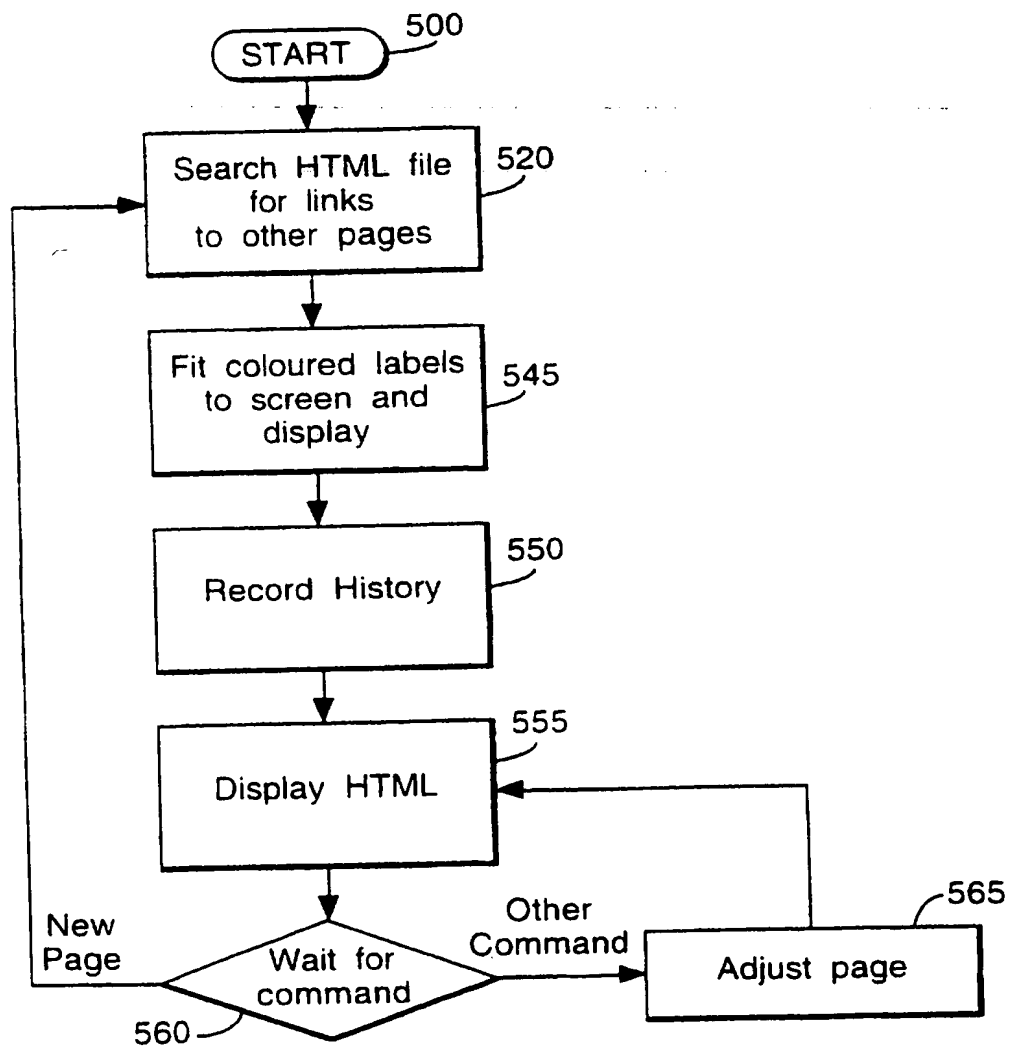


Fig.6.

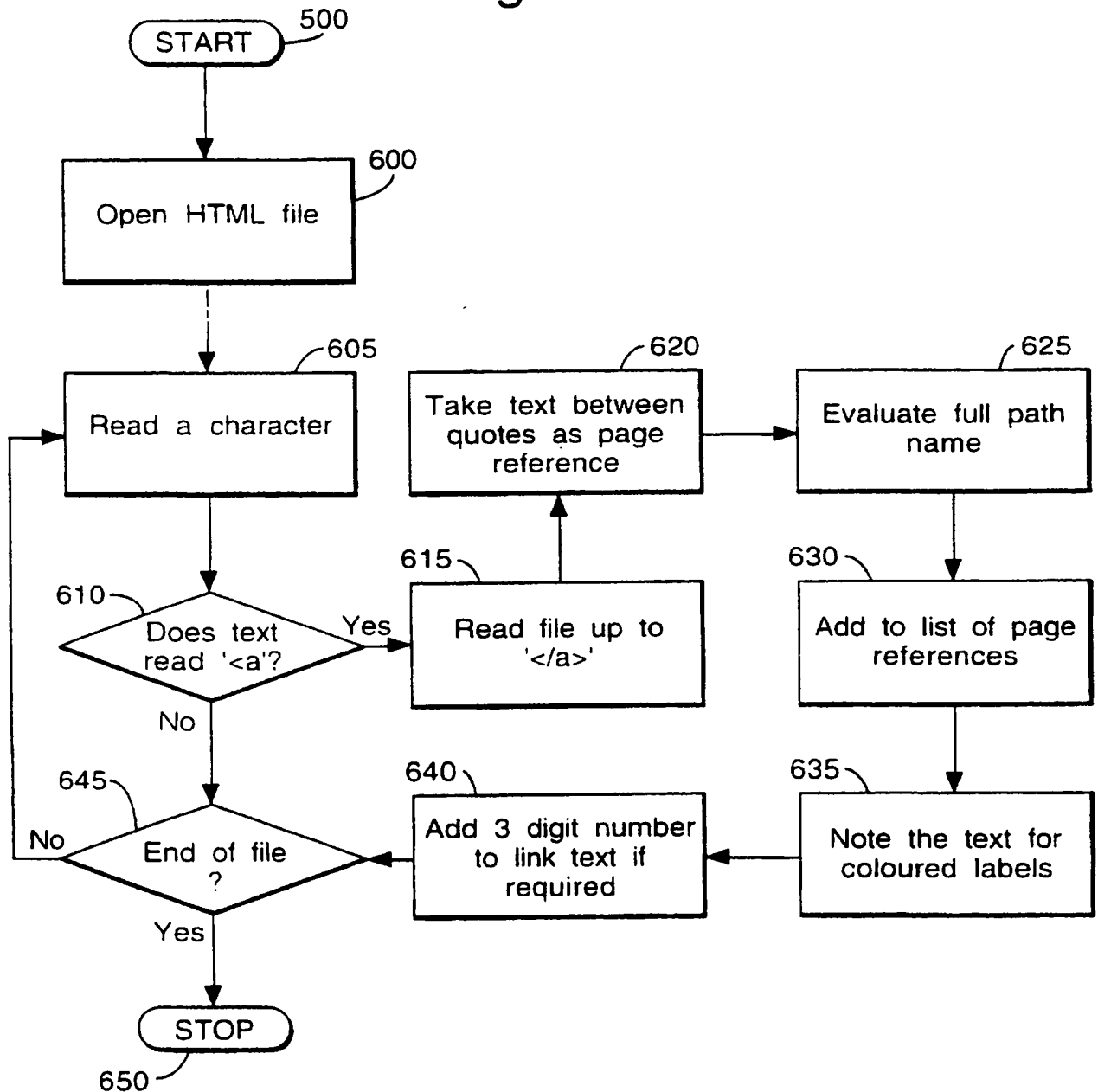


Fig.7.

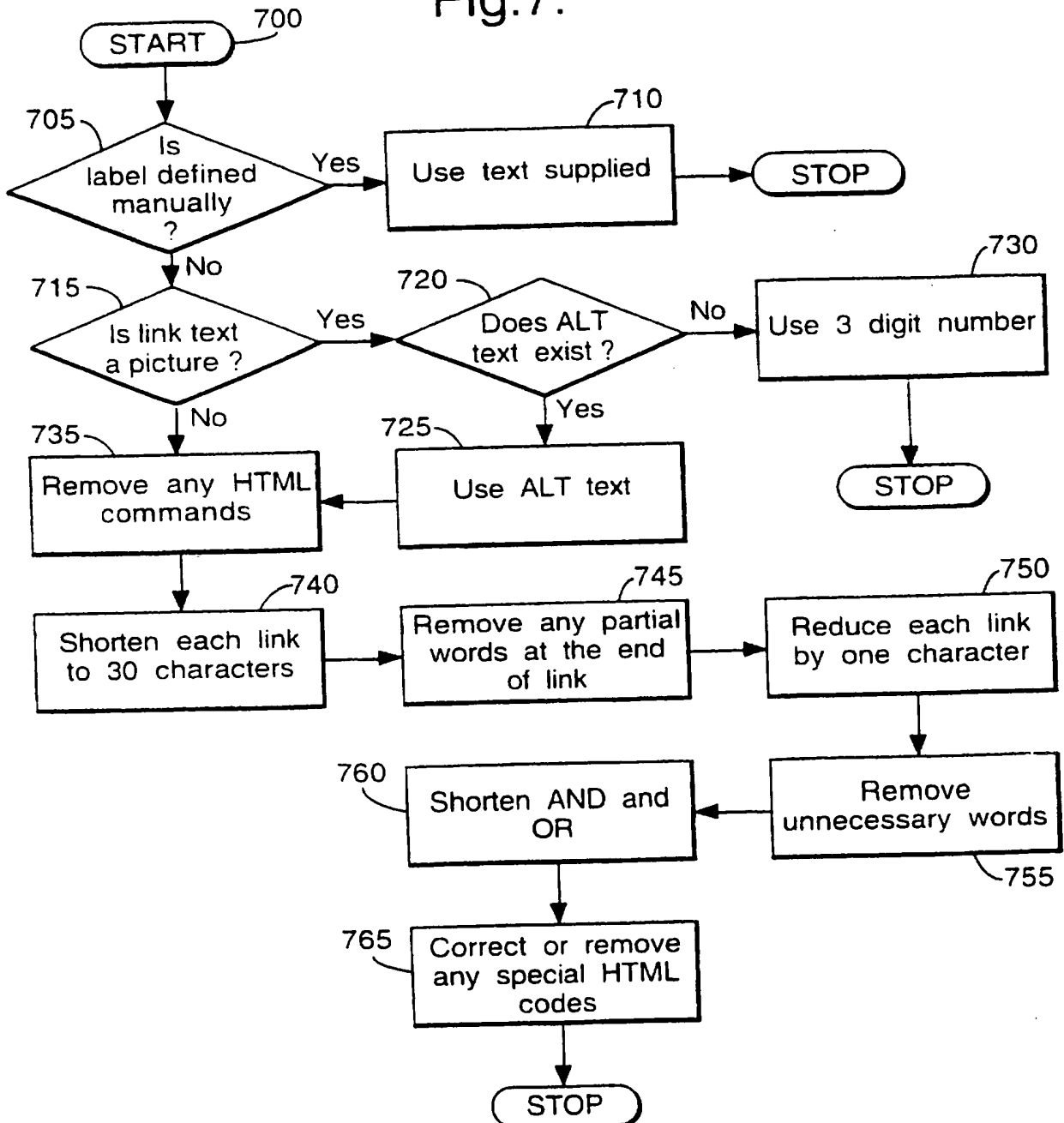


Fig.8.

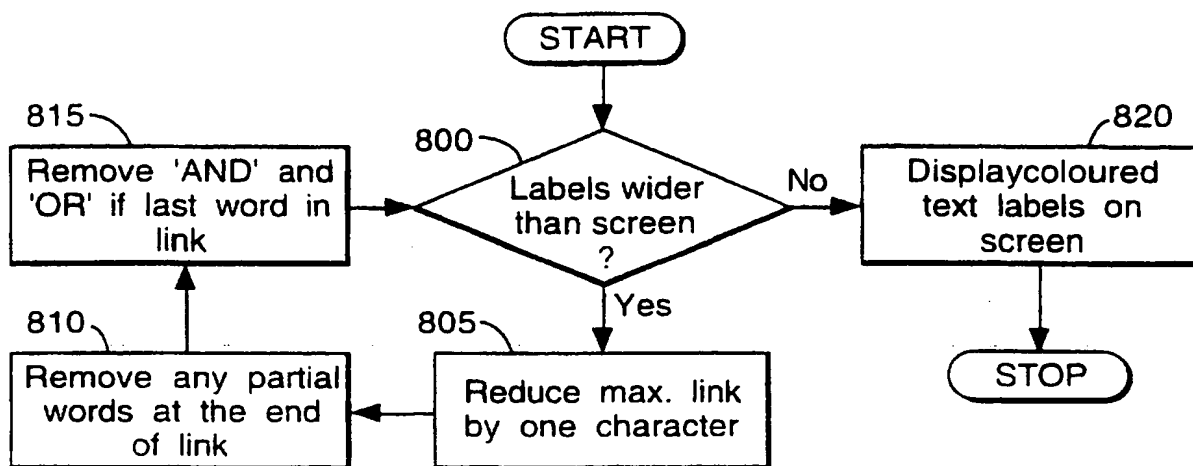
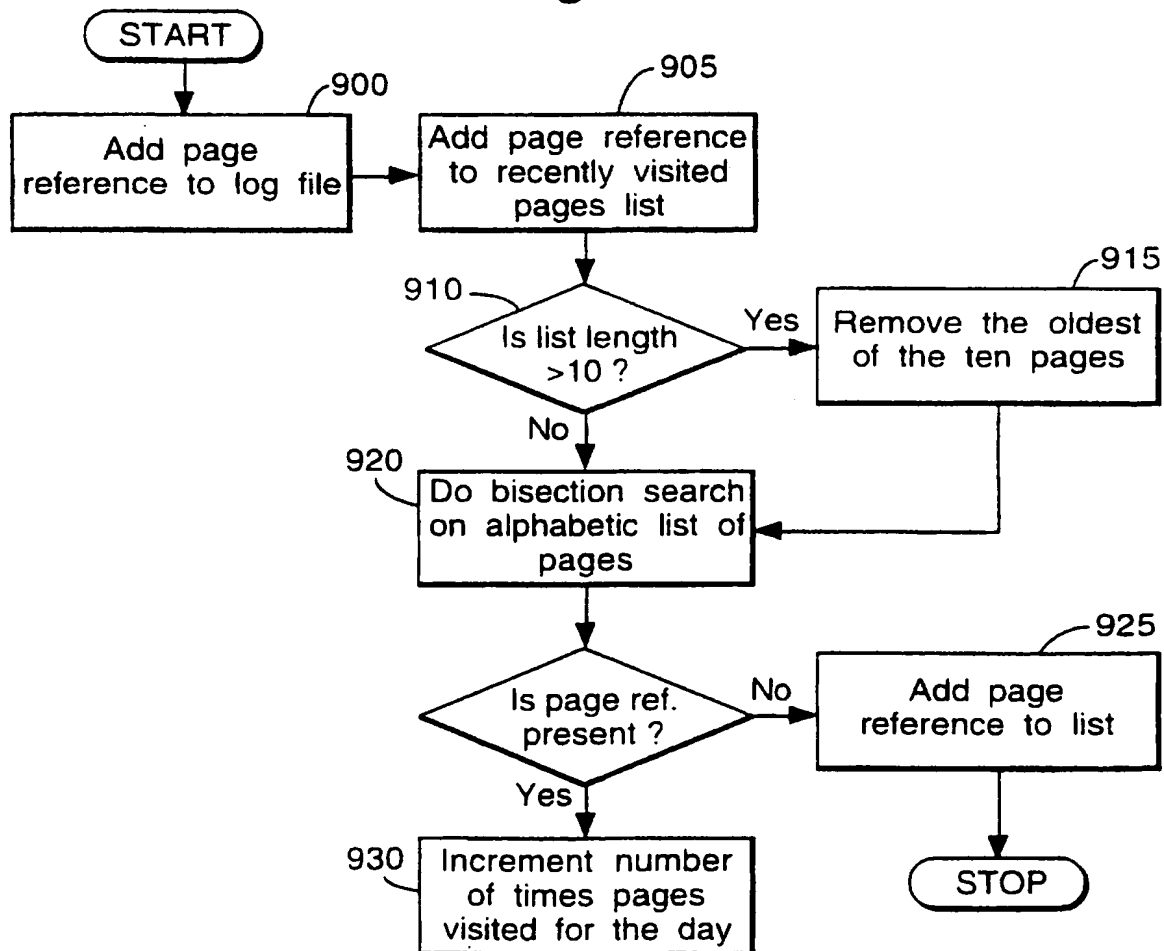


Fig.9.



7/10

Fig.10.

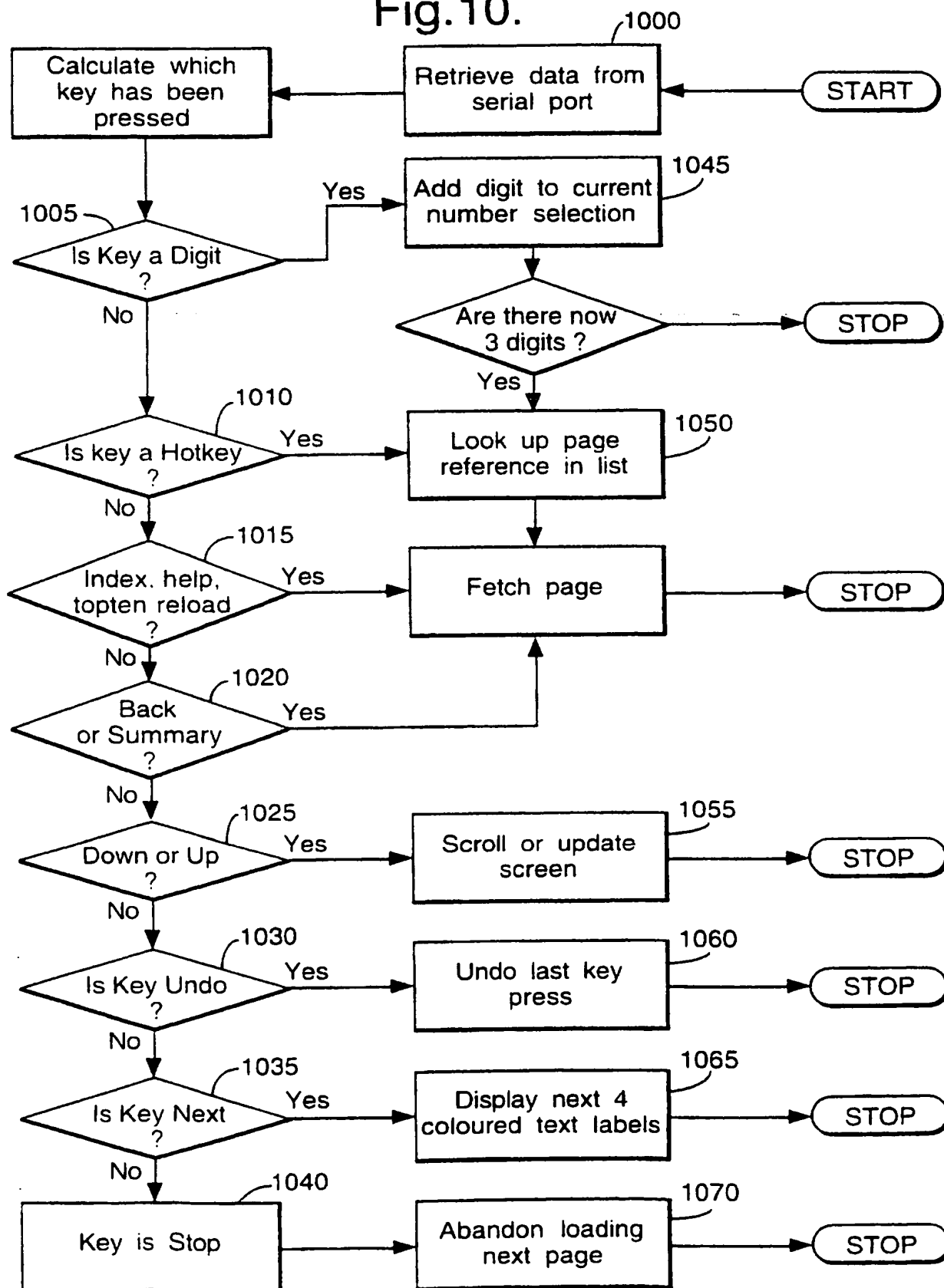


Fig.11.

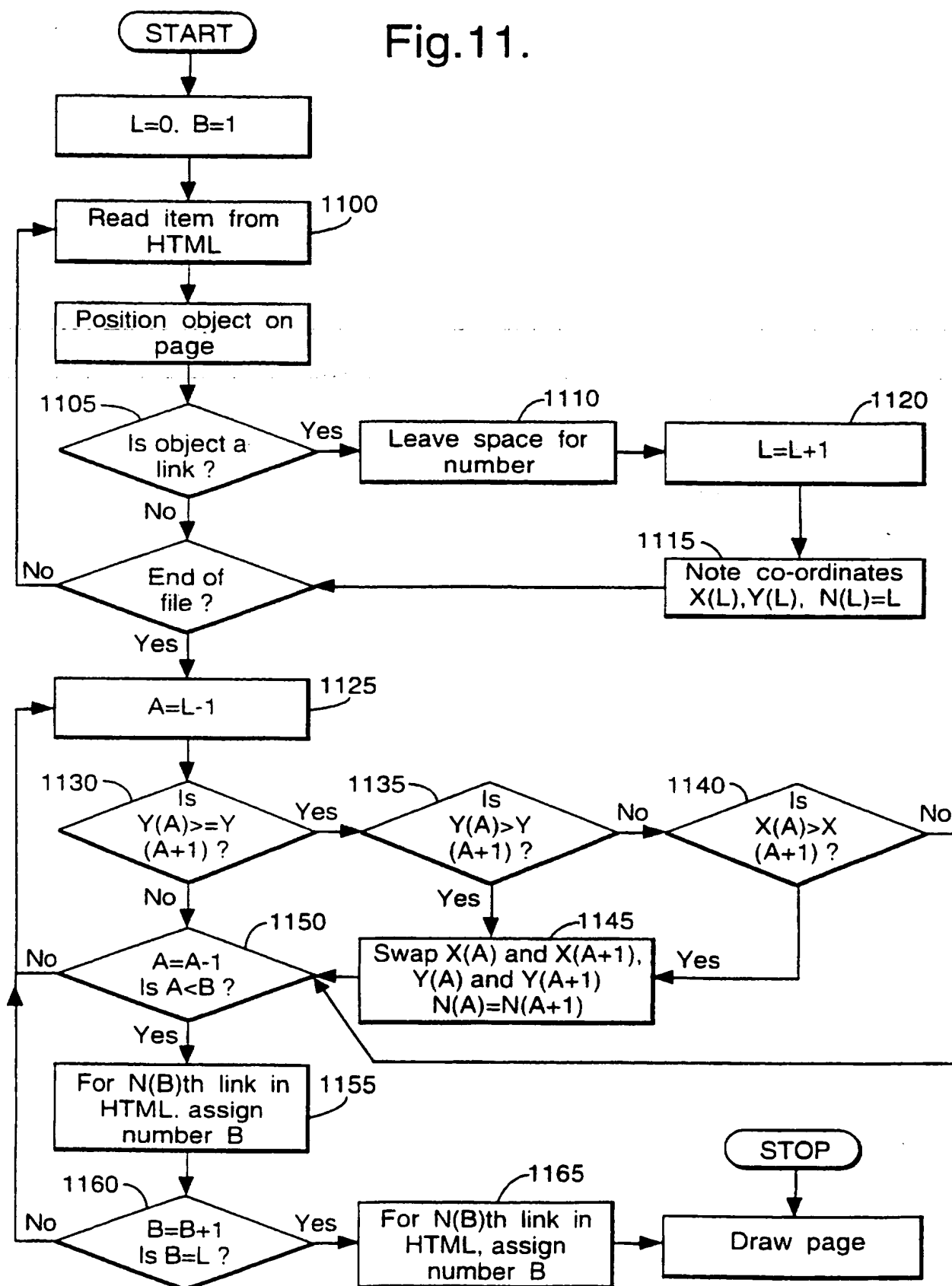


Fig.12.

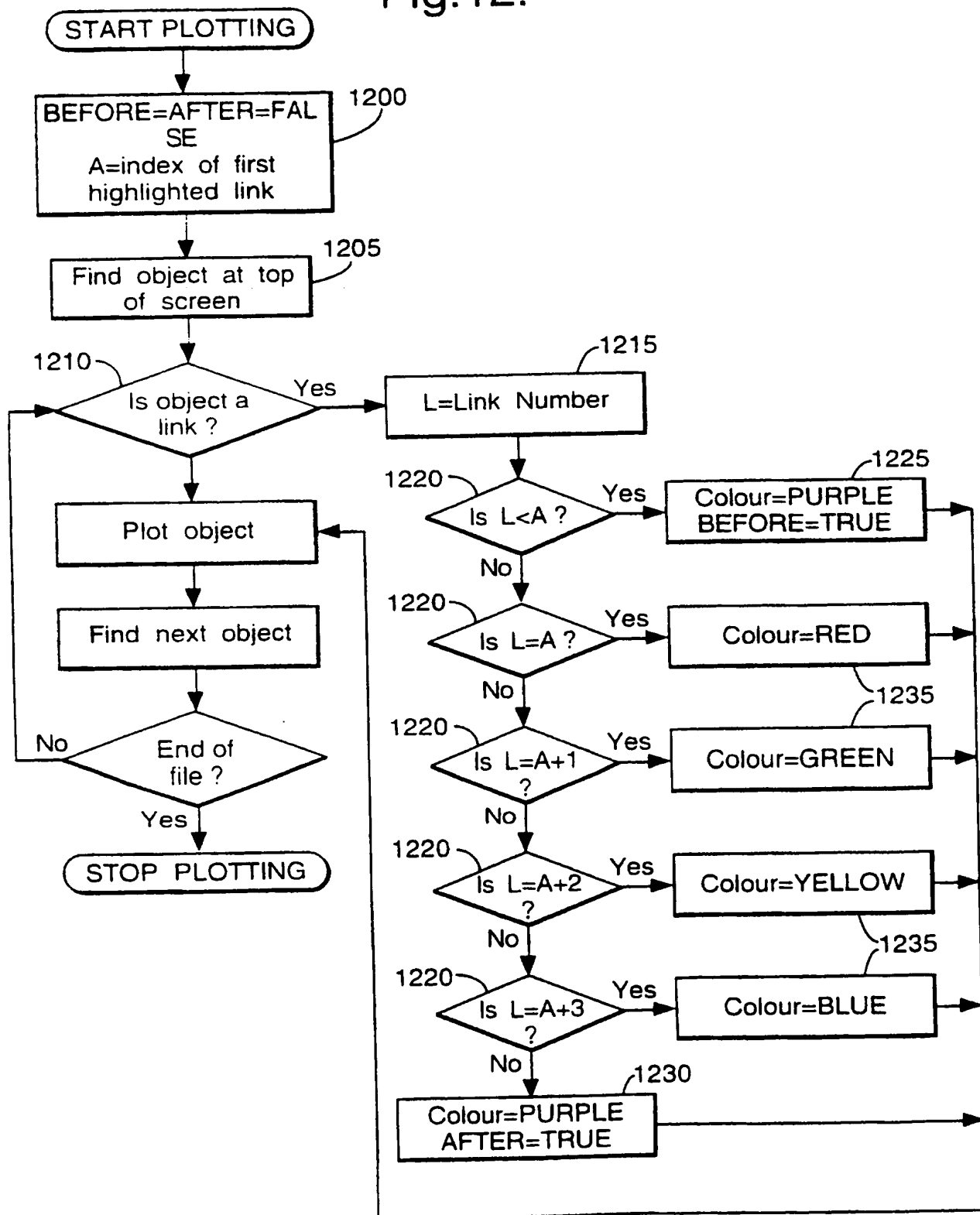


Fig.13.

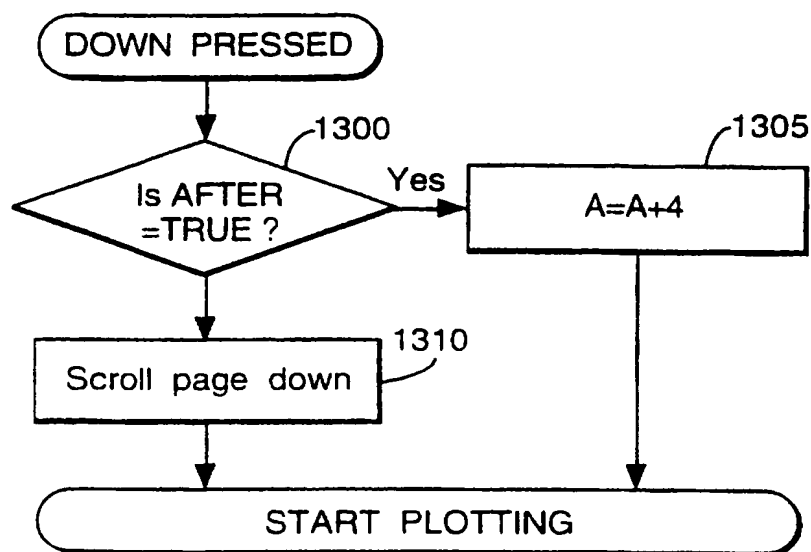
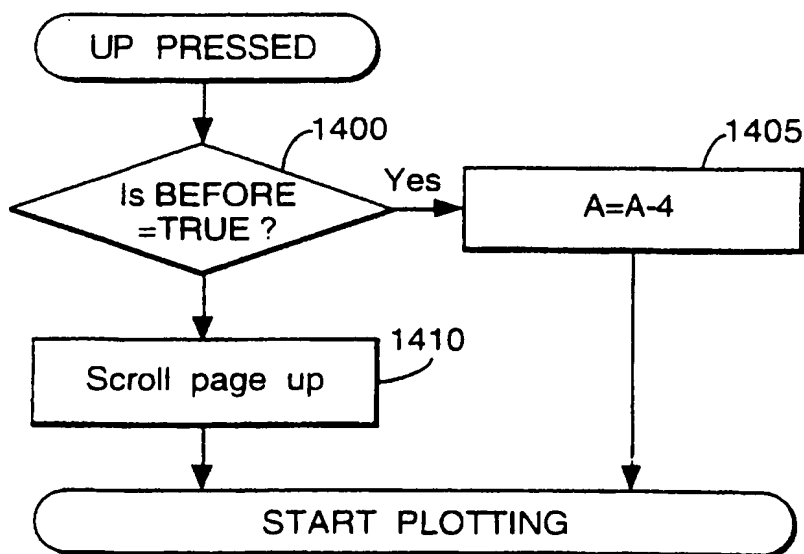


Fig.14.



INTERNATIONAL SEARCH REPORT

International Application No
PCT/GB 97/01627

A. CLASSIFICATION OF SUBJECT MATTER
IPC 6 G06F17/30

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 6 G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	BEHAVIOR RESEARCH METHODS AND INSTRUMENTATION, vol. 27, no. 2, 1 January 1995, pages 200-205, XP000567305 CHU J Y M ET AL: "CREATING A HYPertext MARKUP LANGUAGE DOCUMENT FOR AN INFORMATION SERVER" cited in the application see abstract; figure 1 see page 200, column 2, line 11 - page 201, column 1, line 4 see page 203, column 2, line 29 - page 205, column 2, line 13	1-35
Y	US 5 459 489 A (REDFORD PETER M) 17 October 1995 see abstract see column 1, line 48 - column 2, line 36	1-35

☐ Further documents are listed in the continuation of box C.

☒ Patent family members are listed in annex.

* Special categories of cited documents:

- * "A" document defining the general state of the art which is not considered to be of particular relevance
- * "E" earlier document but published on or after the international filing date
- * "L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
- * "O" document referring to an oral disclosure, use, exhibition or other means
- * "P" document published prior to the international filing date but later than the priority date claimed

- * "T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
- * "X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
- * "Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
- * "&" document member of the same patent family

Date of the actual completion of the international search

16 September 1997

Date of mailing of the international search report

24.09.97

Name and mailing address of the ISA

European Patent Office, P.B. 5818 Patentlaan 2
NL - 2280 HV Rijswijk
Tel. (+31-70) 340-2040, Tx. 31 651 epo nl.
Fax: (+31-70) 340-3016

Authorized officer

Katerbau, R

Information on patent family members

PCT/GB 97/01627

Patent document
cited in search report

Publication date

Patent family member(s)

Publication date

US 5459489 A

17-10-95

US 5339095 A

16-08-94

US 5218771 A

15-06-93

US 5650608 A

22-07-97

Prefetching in World Wide Web

Zheng Wang
Jon Crowcroft

Department of Computer Science, University College London
London WC1E 6BT, United Kingdom

Abstract

This paper considers the use of prefetching in WWW for reducing perceived latency. We first look at a number of basic issues and tradeoff in prefetching. Our analysis shows that, unless the traffic is very light or the prefetching efficiency is very high, statistical prefetching may not necessarily reduce perceived delay. We then present an implementation of deterministic prefetching in a hotlist manager.

1 Introduction

As the World Wide Web (WWW) is now the *de facto* standard interface for interactive information services over the Internet, the perceived latency in WWW interaction is becoming an important issue. Currently, Web users often experience response delay of several seconds or even longer to non-local Web sites. For WWW to be acceptable for general daily use, the response delay has to be substantially improved.

The potential solutions to the problem may lie in the extensive use of caching and prefetching in WWW. Both caching and prefetching exploit the patterns and knowledge in the Web page accesses. Whilst caching exploits the probability of multiple requests to the same Web page, prefetching makes use of the knowledge in a client's requests to multiple Web pages.

In this paper, we focus our attention on the use of prefetching for reducing delay in WWW. We first look at latency and access patterns in WWW interaction, and examine basic issues and tradeoffs in prefetching. We then discuss deterministic client-initiated prefetching in some detail and present an implementation in the HotList Manager.

2 Latency in WWW Interaction

The perceived latency in WWW interaction comes mainly from two sources. At the lower level, the round trip time (RTT) is the most fundamental one. The RTT has two main components: processing latency in end systems, and communication latency over the network. The processing latency depends on the load of the end systems. The communication latency consists of queuing delay and propagation delay. An increase of bandwidth can reduce queuing delay but the propagation delay may not be changed.

The communication protocols decides the number of RTTs required for each interaction thus also affect the perceived latency. In current HTTP implementation, each object in a Web page requires a new TCP connection, hence at least 2 RTTs. The new persistent HTTP can reuse TCP connections so it requires only one TCP setup for each interaction [1].

The latency experienced by users varies considerably depending on the bandwidth available, the propagation delay to the server, the size of the Web page and the server load. To illustrate the delay Web users currently experience, we calculated the delay distribution with 79,430 samples from the client-based traces collected at Boston University during November 1994 to February 1995 [2]. Figure 1 shows the distribution of delay for retrieving an object from non-local servers (i.e., the servers other than the ones on users' LANs). For most objects, the retrieving time is between 0.4-4 seconds. A Web page usually contains a number of embedded objects, thus retrieving time for one web page is several times longer. The distribution of the transmission rates from non-local servers, produced with the same dataset, is shown in Figure 2. It is clear that bandwidth remains the dominant factor in latency.

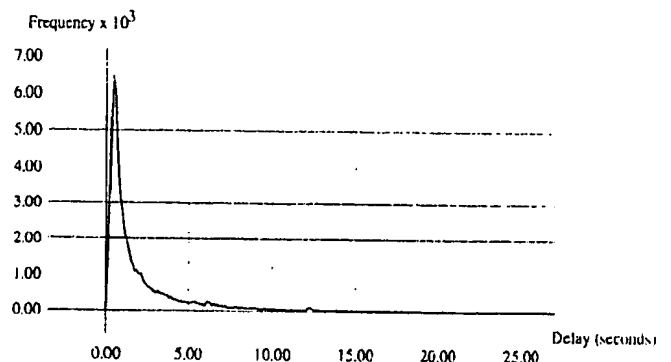


Figure 1: Distribution of Delay for Retrieving an Object from Non-Local Servers

3 Prefetching and Access Patterns

~~Prefetching tries to reduce the perceived latency by looking ahead.~~ If one can anticipate the Web pages that a user will fetch in the future, it is possible to preload these Web pages

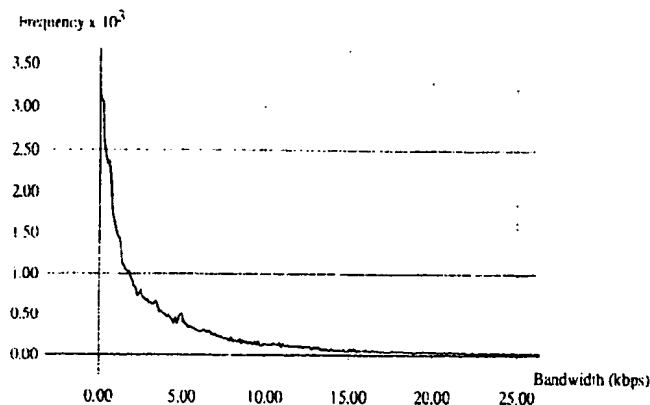


Figure 2: Distribution of Transmission Rates from Non-Local Servers

to the user before they are requested. The perceived latency can then be reduced to that for fetching Web pages from local disks or file servers on local nets. Thus, the effectiveness of prefetching depends on whether there is certain predictability in users' Web page accesses.

The information on access patterns may be derived from servers' access statistics or from clients' configuration. Recent studies on the WWW traffic show that there are considerable interdependencies among consecutive accesses to some Web pages [3]. Such results are not surprising. As WWW is a hypertext based information system, it is natural that many Web pages are closely linked together. The design of a Web page also places some physical constraints on what hyperlinks can be followed from a particular page, and the contents may also provide some strong leads as to the order in which the Web pages should be viewed. Such information may be used as the basis for predicting users' Web page accesses.

Users' personal preferences are also an important factor. The sequence of accesses is ultimately decided by users' individual selection. For regular services such as on-line newspapers, users often know in advance what they want to see. Information on users' future accesses may also be derived from users' planning tools such as diaries or todo lists.

4 Basic Issues in Prefetching

In this section, we examine some important issues in prefetching and discuss the tradeoffs between bandwidth and delay.

Server-Initiated vs Client-Initiated Prefetching

Prefetching may be initiated by clients or by servers. When a Web page is request, a server may anticipate what hyperlinks are likely to be followed, and preload the corresponding Web pages to the client. So, when the user moves to a hyperlink, the corresponding Web pages may already be transferred or being transferred.

For server-initiated prefetching to work, the client has to be prefetching-aware so it can deal with preloaded Web pages correctly. This would require extensions to the current HTTP protocol and modifications to both the client and server software.

A client can also initiate prefetching Web pages based on user's configuration or a user agent can monitor the patterns of past accesses for particular Web pages, and prefetch Web pages on the behalf on the user.

Client-initiated prefetching can be done by individual clients in a way transparent to the servers. The implementation is therefore much simpler.

Server-Initiated vs Client-Initiated Prefetching

The criteria for deciding whether a Web page should be prefetched can be either statistical or deterministic. In a statistical scheme, one can calculate the interdependencies of Web page accesses periodically based on the most recent access logs, and group Web pages with interdependencies higher than a certain threshold for prefetching.

Prefetching can also be configured statically by the users as part of their personalized user interfaces, or even by page designers as part of the content design. For example, a user may configure some sections of an on-line newspaper that are always read with prefetching so that these sections are prefetched when the newspaper's home page is accessed.

4.1 Bandwidth and Delay Tradeoff

Statistical prefetching can potentially have wide applications as the process can be easily automated. However, by its speculative nature, some bandwidth will be wasted, thus it can increase total bandwidth consumption. There is a general tradeoff between bandwidth and latency here. As we reduce the threshold for statistical prefetching, the latency may improve, but at the price of increased bandwidth consumption. A study on FTP shows that the latency can be reduced by 67% for 7-fold increase in bandwidth [4].

Unfortunately, bandwidth is still scarce resources in most networks, particularly over long distance paths. Thus, statistical prefetching has been used with great care to avoid any waste of bandwidth. Note that, although the perceived delay for prefetched files are very low, the retrieving delay for non-prefetched files may actually increase as a result of the extra traffic load caused by prefetching. When traffic is heavy, aggressive prefetching, such as "get all links", may actually increase the average latency of all accesses.

We now look at the tradeoff in some detail. Suppose that P is the hit rate of prefetching (i.e., the probability of a correct prefetch), and D_0 is the average retrieving delay without prefetching. Assuming that the retrieving delay is 0 for prefetched Web pages, and the delay is D_r for non-prefetched ones. So the average delay with prefetching D_n can be written as

$$D_n = P \times 0 + (1 - P) \times D_r = (1 - P) \times D_r$$

To ensure that prefetching reduces the average delay, i.e., $D_n < D_o$, we have

$$D_r/D_o < 1/(1-P)$$

For the sake of simplicity, we assume that the delay D_x and D_o can be calculated based on M/M/1 model, i.e.,

$$D_o = 1/(1-R_o)$$

$$D_r = 1/(1-R_x)$$

where R_o and R_x are respectively the link utilization without and with prefetching. We then get

$$P/((R_x - R_o)/R_o) > R_o/(1 - R_o)$$

We define the efficiency of the prefetching E as the ratio of the hit rate of prefetching and the rate of traffic increase to achieve that hit rate, i.e.,

$$E = P/((R_x - R_o)/R_o) > R_o/(1 - R_o)$$

The above implies that the efficiency of prefetching must be larger than $R_o/(1 - R_o)$, otherwise, the average delay can actually be higher than that without prefetching. The inequality can also be written as

$$R_o < E/(1 + E)$$

If E is known, one can calculate the maximum R_o for statistical prefetching to be useful. Figure 3 and 4 show the feasible regions for E and R_o . It is clear that prefetching is only useful when traffic is very light or prefetching efficiency is very high. For example, for $E = 0.5$ (i.e. for each 1% traffic increase, the prefetch hit rate improves 0.5%), R_o must be smaller than 0.3. For $R_o = 0.8$, E must be larger than 4. This is because when traffic is heavy, very little extra traffic may result in substantial increase in queuing delay. Unless the prefetching efficiency is very high, the extra delay experienced by non-prefetched pages may outweigh the decrease in the delay of prefetched pages.

Prefetching Through Caching Proxies

When prefetching is done through caching proxies, the prefetched Web pages are also cached at the proxies. This, in fact, has a number of advantages over prefetching Web pages directly to the client's machines. First, the prefetched Web pages on the proxies can be cached pages to other users who use the same proxies. Second, if the client's local cache is on a file server, to get a Web page from the local cache involves transmission over local nets anyway. In such cases, using caching proxies also on local nets has no extra communication costs. Finally, the caching format on proxies is not browser-specific, so the prefetching may be implemented in a standalone program outside the browsers. In section 5, we will present an implementation of prefetching in a hotlist manager.

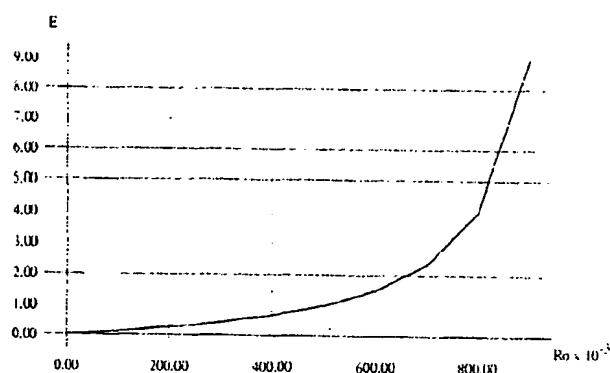


Figure 3: Feasible Region for E (Area above solid line $E > R_o/(1 - R_o)$)

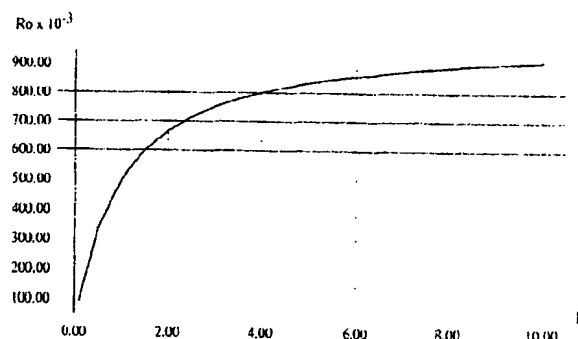


Figure 4: Feasible Region for R_o (Area above solid line $R_o < E/(1 + E)$)

5 Deterministic Client-Initiated Prefetching

Deterministic prefetching is the most conservative type as it often has little or no bandwidth overhead, although its scope of use is limited. Nevertheless, when users know what needs to be prefetched, it can reduce perceived latency, and to some extent, even ease congestion at very little cost.

In deterministic client-initiated prefetching, prefetching is configured statically by the users. It can be implemented as part of the browser or simply as an add-on without changes to client and server software.

In this section, we discuss some potential use of the deterministic client-initiated prefetching in detail.

Batch Prefetching

There are many Web pages that are read on a regular basis, such as on-line newspapers, weekly work reports etc. They can be prefetched during the less busy period (e.g. early morning). For large Web pages, such as papers with large graphics, it may also make sense to fetch them in a batch

mode and view them later on. Such prefetching, to some extent, is similar to mirroring but batch prefetching is more flexible as it does not require any central administration. It can be done through cache proxies so that all prefetching requests are naturally aggregated at the proxies. Batch prefetching can also potentially allow multicast to be used in distributing some large popular files or software.

Start-Up Prefetching

Prefetching can also be carried out when a browser is started. A set of pages users need to look at that day may be prefetched in the background. It can be integrated with planning tools so that a to-do Web Page is constructed each day for the users and corresponding Web pages are prefetched at the start-up for later viewing.

Pipelining with Prefetching

The current model for navigating is a series of "click, fetch and view" operation. As a user usually spends some time (seconds or minutes) on a page, we can potentially pipeline the operation by fetching the next page while the user is looking at the current one. This is particularly useful for some information services, such as on-line newspapers, stock market prices and headline tracking services where users can easily specify the sequence of pages to be viewed.

6 Implementation

In this section, we present details of a UNIX implementation of deterministic client-initiated prefetching in a program called Coolist.

Coolist is a standalone hostlist manager implemented in Java. It has a number of useful functions such as proxy selection, personal search engine and prefetching. Coolist can be used in conjunction with Netscape, Mosaic and any Web browsers which support client APIs and caching proxy.

Coolist sits between the browser and caching proxy (Figure 5), and acts as a personal proxy. Under most circumstances, Coolist simply forwards traffic to and from the caching proxy. However, as Coolist can intercept any traffic between the browser and the proxy, it can also work on the requests and responses in various ways.

Although Coolist can prefetch most web pages directly by itself, it does not work with Web sites that require passwords, cookie or transparent redirection. In our implementation, Coolist uses client APIs [5] to control browsers to do prefetching. When Coolist needs to fetch a web page, it issues a client API command to the browser to open the web page specified.

For example, with NCSA Mosaic TCP/IP CCI, Coolist sends the following message a Mosaic browser:

```
GET URL <http://www.ncsa.uiuc.edu/SDG/
Software/XMosaic/CCI/cci-api.html> CURRENT
```

The browser receiving the message will open the specified URL and display in the current window. Invoking Netscape as:

```
netscape -remote 'openURL(http://home.netscape.com/
newsref/std/x-remote.html)'
```

has the same effect on Netscape browsers.

When doing batch prefetching, Coolist forks out a cron job with a line mode browser to fetch the web pages at specified time. In our implementation, we choose Lynx [6] (with the line mode option) since it is available on most Unix systems.

Although prefetched pages can be cached in Coolist, we use the caching proxy as the store for prefetched pages. This simplifies the design of Coolist and that prefetched pages can also be shared by other users connected to the same caching proxy.

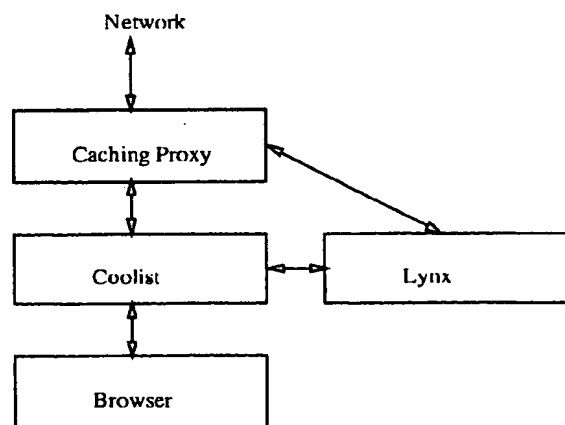


Figure 5: Implementation of Client-Initiated Prefetching in Coolist

Start-up prefetching is configured with a file called `.todo` in user's home directory. When Coolist is invoked, it starts a low priority thread to prefetching all web pages listed in the `.todo` file. Web pages can be grouped together for pipelining prefetching which appears as a single entry in the bookmark.

7 Summary

In this paper, we considered the use of prefetching in WWW for reducing perceived latency. We examined a number of issues and tradeoffs. Deterministic client-initiated prefetching was discussed in some detail and an implementation based Coolist was presented.

References

- [1] Jeffrey Mogul, *The Case for Persistent-Connection HTTP*, in Proceedings of ACM SIGCOMM'95, Oct. 1995.

- [2] Carlos Cunha, Azer Bestavros, and Mark Crovella, *Characteristics of WWW Client-based Traces*, Technical Report TR-95-010, Boston University, Boston, MA 02215, April, 1995.
- [3] Azer Bestavros, *Using speculation to reduce server load and service time on the WWW*, in Proceedings of CIKM'95: The Fourth ACM International Conference on Information and Knowledge Management, Baltimore, Maryland, November 1995.
- [4] J.D. Touch and D.J. Farber, *An Experiment in Latency Reduction*, in Proceedings of IEEE INFOCOM'94, Toronto, June 1994.
- [5] Tom Magliery, Briand Sanderson, *Writing Web software through client APIs*, In the Tutorial Notes of 4th World Wide Web Conference, Boston USA, Nov 1995
- [6] Lynx, *Lynx On-Line Documents*, (<http://ftp2.cc.ukans.edu/pub/lynx/>), 1995.